VŠB - Technická univerzita Ostrava

Fakulta elektrotechniky a informatiky

Katedra informatiky

Automatizace nastavení serverů za pomoci NSH skriptování v prostředí BMC Server Automation

Automation Server Settings by Using NSH Scripting in the BMC Server Automation Environment

2014 Lumír Chovanec

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Lumír Chovanec**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: Automatizace nastavení serverů za pomoci NSH skriptování v prostředí BMC Server Automation
Automation Server Settings by Using NSH Scripting in the BMC Server Automation Environment

Zásady pro vypracování:

Cílem práce je vytvoření několika konkrétních NSH skriptů a jobů v prostředí BSA (BMC Server Automation), které budou sloužit k automatizaci nastavení monitoringu serverové infrastruktury.

1. Nastudujte a popište prostředí produktu BMC Server Automation.
2. Sepište požadavky na automatizaci nastavení monitoringu IT infrastruktury.
3. Vytvořte NSH skripty pro změny nastavení monitoringu IT infrastruktury.
4. Zautomatizujte celý proces pomocí úloh (jobs).
5. Otestujte nasazení a porovnejte jeho reálné výsledky s teoretickými předpoklady na použité IT infrastruktuře (WIN + UNIX).
6. Vyhodnoťte výsledky testů a navrhněte další budoucí rozšíření.

Seznam doporučené odborné literatury:

[1] BMC SOFTWARE. BMC BladeLogic Network Shell Command: Reference manual. Version 8.1.00. HOUSTON, USA, 2011.
[2] STOLZ, Annette. Microsoft Windows Server 2003 skripty: velká kniha řešení. Vyd. 1. Překlad David Čepička. Brno: Computer Press, 2007, 699 s. ISBN 978-80-251-1668-5.
[3] SHAH, Steve. Administrace systému Linux: překlad čtvrtého vydání. 1. vyd. Praha: Grada, 2007, 426 s. ISBN 978-80-247-1694-7.
[4] KOLEKTIV AUTORŮ. Automatizace a automatizační technika 1: systémové pojetí automatizace. 1. vyd. Brno: Computer Press, 2012, 217 s. ISBN 978-80-251-3628-7

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Richard Biječek**

Konzultant bakalářské práce: Ing. Pavel Moravec, Ph.D.

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014

doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení

„Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal."

Tato práce obsahuje citlivá firemní data ve formě NSH skriptů, které jsou umístěny v neveřejné části této bakalářské práce na samostatné CD příloze.

V Ostravě dne 4. května 2014

Lumír Chovanec

Vlastnoruční podpis autora

Abstrakt

Tato práce vyhodnocuje význam automatizace nastavení monitoringu serverové infrastruktury. Práce předkládá způsob analýzy monitorovaných událostí a vyhodnocuje rozdíl mezi automatickým a manuálním způsobem nastavování monitoringu. V prostředí BMC Server Automation byly za pomoci skriptovacího jazyka NSH vytvořeny skripty, sloužící k univerzálnímu nastavení nové konfigurace monitoringu PATROL Agenta na serveru. Tyto skripty byly následně za pomoci jobů spouštěny na všech monitorovaných serverech. Tím bylo dosaženo významné úspory pracovní síly k provedení změny nastavení a zajištění spolehlivosti nového nastavení. Výsledek automatického nastavení byl vyhodnocen a po třech měsících od aplikace nového nastavení došlo k redukci nepotřebných informací z monitoringu o 70 %.

Klíčová slova

Monitoring, NSH skriptovací jazyk, PATROL Agent, BMC Server Automation,

IT infrastruktura, server, ITIL, Event management, Incident management, Root Cause

Abstract

This paper evaluates the significance of automated settings of the server infrastructure monitoring. A mode of analysis of monitored events is presented, and the difference between automatic and manual methods of setting the monitoring is evaluated. In a BMC Server Automation environment, the scripts were created using a NSH scripting language, which serves for universal setting of a new configuration of a PATROL Agent monitoring on a server. Using jobs, those scripts were subsequently run on all monitored servers, thereby significantly reducing the manpower needed to carry out the setting adjustments, and ensure the reliability of a new setting. The result of automatic settings was evaluated: in three months after applying new settings, a 70 % reduction of unneeded monitoring information was achieved.

Keywords

Monitoring, NSH scripting language, PATROL Agent, BMC Server Automation,

IT infrastructure, server, ITIL, Event management, Incident management, Root Cause

Seznam použitých symbolů a zkratek

AD – Active directory, služba systému WINDOWS
ADMX – jazyk na bázi XML vyvinutý společností Microsoft
BEM – BMC Event management, profesionální produkt společnosti BMC pro získávání, evidenci a analýzu událostí na serverech
BMC – společnost vyvíjející Software, 2101 CityWest Blvd., Houston, Texas 77042
BSA – BMC Server Automation, profesionální produkt splečnosti BMC pro automatizaci a správu rozsáhlých IT systémů obsahující řádově tisíce serverů
CPU – Central Processing Unit, procesorová jednotka počítače provádějící výpočetní operace
CSV – Comma-separated values, souborový formát využívaný pro ukládání tabulkových dat
DHCP – Dynamic Host Configuration Protocol, protokol pro dynamické přidělení IP adresy
GPO – Group Policy Objects, hromadné politiky systému Windows
HDD – Hard Disk, pevný disk
HW – Hardware, veškeré fyzické komponenty počítače
ICMP – Echo request, požadavek na odezvu, způsob ověření dostupnosti zařízení s IP adresou
IIS – Internetová informační služba
IP – Internet Protocol, nejpoužívanější protokol pro komunikaci na internetové vrstvě
IT – Information Technology, informační technologie
ITIL – The Information Technology Infrastructure Library, mezinárodní standard pro ITSM
ITSM – IT service management, management pro zajištění kvality poskytovaných IT služeb
KM – Knowledge Modules, inteligentní modul zajišťující monitoring konkrétní komponenty serveru jak HW tak SW. Jedná se tedy především o programový kód načítající stav monitorované komponenty, například vytížení procesoru serveru
NIC - Network Interface Card – síťová karta počítače (hardware)
NSH – Network Shell Command, skriptovací jazyk na principu UNIX sh
NTFS – New Technology File System, souborový systém vyvinutý společností Microsoft
OS – Operation System, operační systém (např.: Windows, Unix, Linux, Android)
PID – Process Identifier, identifikátor procesu běžícího na serveru
RAM – Random Access Memory, dočasná paměť počítače
RSCD agent – agent sloužící ke komunikaci a aplikaci NSH skriptů na serveru
SLA – Service Level Agreement, smlouva mezi poskytovatelem IT služby a zákazníkem
SQL – standardizovaný dotazovací jazyk používaný pro práci s daty v relačních databázích
SW – Software, programové vybavení
OU – Organization Unit, organizační jednotky active directory v systému WINDOWS
VHD – Virtual Disk, virtuální disk specifikace společnosti Windows pro dočasné úložiště
WBADMIN – služba systému Windows pro zálohování systémových souborů
XML – Extensible Markup Language, velmi rozšířený jazyk pro serializaci strukturovaných dat

Obsah

ÚVOD

Při volbě tématu bakalářské práce jsem se rozhodl využít své praktické zkušenosti při správě IT infrastruktury v jedné z IT firem působících přímo v Ostravě. Jako systémový specialista a problem manager musím denně analyzovat desítky případů a rozhoduji o správnosti postupu řešení technických příčin problémů vznikajících v rozsáhlé IT infrastruktuře. Vždy se jedná o nutnost poměrně rychle vyhodnotit, zda může mít daná závada kritický dopad na funkci sytému. Toto rozhodování může značně usnadnit kvalitně nastavený monitoring infrastruktury. Během své osmileté praxe v tomto oboru činnosti jsem dospěl k závěru, že značné množství zaregistrovaných chybových zpráv, lze považovat v určitých situacích za nepodstatné. Moje profesionální znalosti mi umožnily se během posledních tří let soustředit převážně na analýzu tohoto monitoringu a jeho optimalizaci pro správu windows serverů.

Má práce si klade za cíl dosáhnout zvýšení kvality poskytovaných služeb a snížení náročnosti a chybovosti při implementaci změn nastavení monitoringu serverové infrastruktury. V úvodních částech této bakalářské práce se zaměřuji na popis analýzy, která musí předcházet každému požadavku na změnu monitoringu. Právě tato část je zásadním předpokladem následného úspěchu či neúspěchu provedené změny monitoringu. Změna samotná totiž nemusí mít vždy jen pozitivní dopad. Především je nutné brát ohledy na skutečnost, že primárním cílem je zajistit zákazníkovi spolehlivou a fungující infrastrukturu. Na straně druhé je tu oprávněný požadavek na ulehčení monotónní práce při každodenních náročných technických analýzách a předcházení chybám v úsudku, tedy vlivu lidského faktoru. Právě lidský úsudek je to, co stroje zatím nahradit nedokážou. Ale opakující se exaktně definované postupy a činnosti nahradit dovedou.

Díky implementaci profesionálního produktu BMC Server Automation se naskytla příležitost, kterou jsem se rozhodl plně využít. I přes počáteční nezájem ze strany vedení společnosti se mi podařilo za podpory svých kolegů prosadit myšlenku hromadných změn nastavení monitoringu. To se dosud jinými metodami nedařilo realizovat. Dle vlastní koncepce jsem zformuloval novou metodu analýzy a hromadného způsobu nastavování PATROL Agentů. Nahrazením monotónních činností dojde k vyloučení chyb při nastavování monitoringu a rovněž k úspoře práce při nadbytečných analýzách. Tato bakalářská práce popisuje všechny fáze od analýzy dat, návrhu nového nastavení, vývoje skriptů až po jejich nasazení a vyhodnocení efektu. Popisuje také samotný produkt firmy BMC tak, aby se každý mohl rozhodnout, zda je pro jeho potřeby vyhovující či nikoliv.

1 MONITORING IT INFRASTRUKTURY

Monitoring IT infrastruktury je nezbytnou součástí chodu každé společnosti poskytující IT služby. Současné požadavky na výpočetní výkon, stabilitu a rychlost aplikací provozovaných na serverech již výrazně překročily možnosti správy za pomoci výhradně lidských zdrojů a namátkové kontroly stavu serverů, logů, dostupného místa na diskových jednotkách, vytížení a dalších parametrů.

1.1 Teoretické a praktické předpoklady monitoringu

Teoreticky není potřeba mít monitoring aplikován a lze se pouze spoléhat na zpětnou vazbu zákazníka, který reportuje nedostupnost systému. Takové služby by však dnes poněkud zaostávaly za očekáváními zákazníků. Zákazníci vyžadují služby v rozsahu garantovaném servisní smlouvou. Dle ITIL (Bucksteek, 2011) se jedná o Service Level Agreement (SLA). Například si lze se zákazníkem dohodnout dostupnost bankovní aplikace ve výši 99,98 %. Dohledová centra pro takto vysoké požadavky na dostupnost však nelze provozovat jinak než za pomoci velmi sofistikovaných systémů dohledu nad funkčností celé IT infrastruktury. Technologie současnosti již nemohou být závislé pouze na dozoru operátora. Výpočetní technika jako celek je natolik komplikovaný systém, že schopnosti člověka obecně již dávno neumožňují 100% spolehlivost IT infrastruktury. Není od věci srovnat průměrně velké datacentrum s atomovou elektrárnou, kde nikdo o nutnosti dohledového centra ani vteřinu nepochybuje. Velké množství poskytovaných IT služeb je pro zákazníka neméně kritických. Východiskem k zajištění požadované dostupnosti IT služeb je tedy jednoznačně potřeba monitoringu. V podstatě v jakémkoliv IT odvětví se dnes naštěstí lze spolehnout na technologie monitorující samy sebe. Tedy ač to může znít neuvěřitelně, počítače do jisté míry již opravdu dokáží kontrolovat samy sebe a případně spouštět samoopravné mechanismy odstraňující nutnost zásahu lidské ruky. Autorem všech těchto technologií je a ještě dlouho bude člověk. A je jen na něm, jak sofistikované systémy dokáže vytvořit. Základním předpokladem požadované funkce jakéhokoliv systému je však zjištění jeho aktuálního stavu. Tedy monitoring IT infrastruktury.

1.2 Nástroje pro monitoring serverů

První etapou k dosažení excelentní dostupnosti služeb pro zákazníky je instalace nějaké formy monitoringu. Monitoring serverové infrastruktury lze provádět desítkami různých způsobů. Základní rozdělení je na monitoring za pomocí agentů a bez nich. V obou případech je vyžadován přenos shromažďovaných informací pomocí různých protokolů z jednotlivých serverů k centrálnímu zpracování. Zde jsou monitorovaná data jednak vyhodnocována a jednak je na tyto výstupy z monitoringu nějakým způsobem reagováno.

1.2.1 Pasivní monitoring

Základní monitoring poskytuje i jednoduchý ping (ICMP požadavek), který zjistí, zda-li je server „na živu“ a otestuje jeho základní dostupnost. Tento způsob lze však opět použít pouze k základnímu testu průchodnosti síťové cesty mezi dvěma body. Nic nám však už neřekne, zda zařízení na druhé straně je schopno rovněž poskytovat požadovanou službu. Tedy zda je dostupná požadovaná funkce systému a nikoliv jen, zda je otevřená síťová cesta mezi klientem a serverem.

Příklady toho, co lze monitorovat:

- Ping (server je dostupný/nedostupný)
- Webserver (webserver je dostupný/nedostupný)
- Rychlost načtení stránek
- Funkčnost SQL databáze
- Přístupnost webových stránek

Aplikace instalované na serveru však již vyžadují daleko sofistikovanější způsob monitoringu. Protože aplikací jsou tisíce, tak i monitoring takových aplikací je možno provádět pravděpodobně tisíci různými způsoby. Například lze pouze opisovat stav z aplikačních logů serveru, kde aplikace sama dává o sobě informace o svém stavu. Výpis 1 představuje ukázku záznamu o navázání spojení aplikace Outlook s Microsoft Exchange serverem.

```
Source:       Outlook
Date:         28.3.2014 14:53:13
Event ID:     26
.....
Connection to Microsoft Exchange has been restored.
Event Xml:
<Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
  <System>
    <Provider Name="Outlook" />
    <EventID Qualifiers="16384">26</EventID>
    <Level>4</Level>
    <Task>0</Task>
    <Keywords>0x80000000000000</Keywords>
    <TimeCreated SystemTime="2014-03-28T13:53:13.000000000Z" />
    <EventRecordID>166182</EventRecordID>
    <Channel>Application</Channel>
    <Computer> computername.domain.com</Computer>
    <Security />
  </System>
.....
```

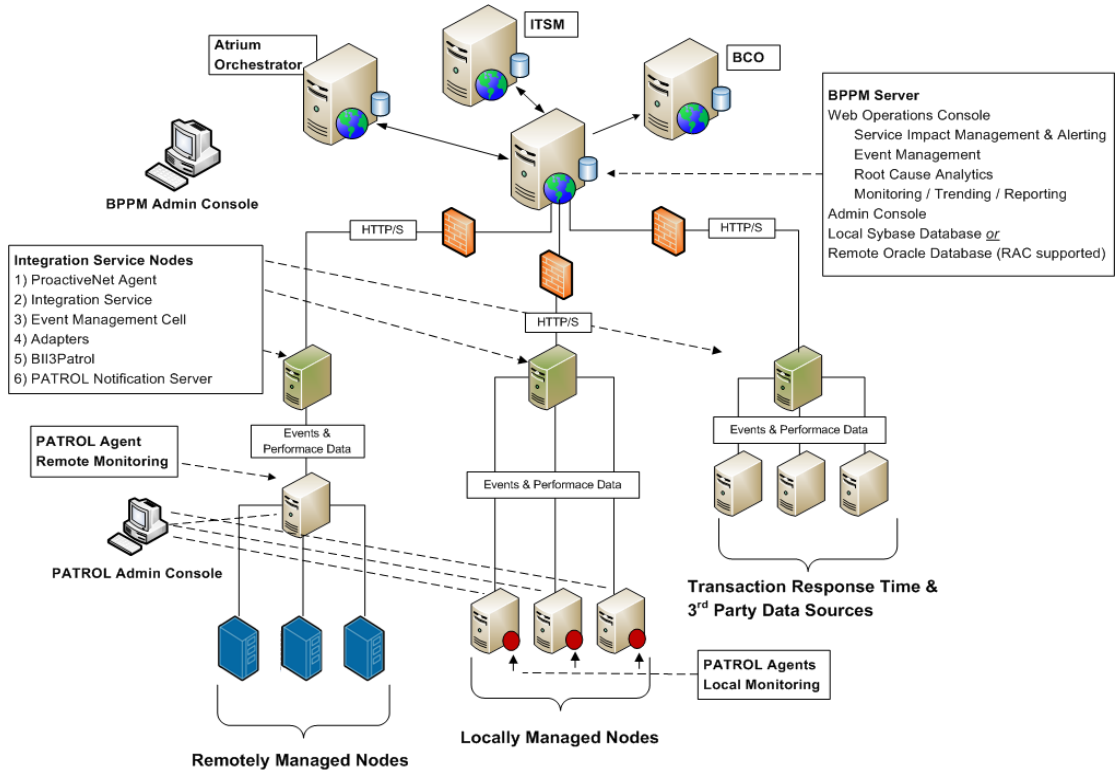
Výpis 1: Záznam chybové zprávy Exchange serveru vyexportovaný do formátu XML

V případě zachycení chyby úrovně Error pak lze nějakým způsobem zareagovat. Na první pohled je však patrné, že tento způsob je velmi neefektivní, neboť na serveru se registruje desítky takovýchto událostí každou hodinu. Zejména nelze získat jasnou představu o souvislostech mezi jednotlivými zprávami. Bez další analýzy nelze rozhodnout, která událost je důležitá a která ne. Proto je v rozsáhlejších sítích vhodnější profesionální nástroj, který tuto práci udělá za člověka a „dostatečně kvalifikovaně“ rozhodne o stavu systému a případné nutnosti zásahu a nápravě chyby, nebo na přítomnosti chyby upozorní dohledové centrum.

1.2.2 PATROL Agent firmy BMC

Monitoring pomocí produktů firmy BMC patří k profesionálním nástrojům pro monitorování stavu serverů a jeho služeb. Na obrázku 1 je kompletní schéma monitoringu tak, jak jej doporučuje nasadit firma BMC. Na monitorovaných serverech je buď lokálně naistalován aktivní PATROL Agent (dole uprostřed) nebo je sběr dat zajištěn sběrem prostřednictvím Remote Monitoringu (vlevo dole) z centrálního bodu směrem k serveru.

Do centrálního místa, kde se monitorovaná data shromažďují, pak mají přístup jednak automatizační nástroje (Atrium Orchestrator, Reporting, Event Management, tiketovací nástroje ITSM), tak dohledové centrum prostřednictvím webového rozhraní. Odsud je pak možno spravovat nastavení jednotlivých agentů, a to jak jednotlivě, tak hromadně, prostřednictvím dalších aplikací, které budou popsány samostatně v další kapitole.



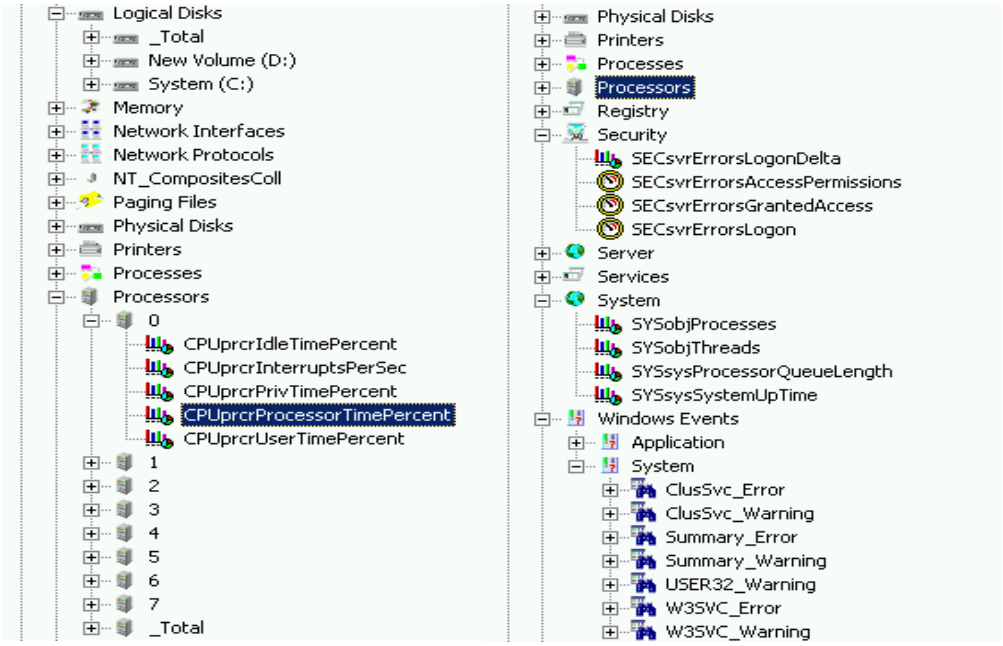
Obrázek 1: Doporučená konfigurace prostředí monitoringu firmy BMC

PATROL Agent pracuje přímo na serveru zcela samostatně. Jeho úkolem je aktivně shromažďovat informace o zdraví serveru a všech jeho komponent. Jedná se například o jednotlivé HW komponenty, jako je CPU, HDD, RAM. Dále o součásti operačního systému, tedy stav jeho služeb, jobů, logů, až po samotné aplikace jako je SQL, IIS. Nelze opomenout zprávy monitorující přenos dat na síťové kartě, průběhu záloh, antivirové kontrole, stav plánovaných úloh a další.

Tato úloha není pro Patro Agenta nijak triviální, neboť je nutná detailní znalost chodu jednotlivých HW a SW komponent, které samy o sobě mohou generovat tisíce chybových stavů. V součtu může být již samotný úkol optimalizace monitoringu jednoho serveru komplikovaný i pro vyškoleného specialistu, který nad jeho konfigurací může strávit desítky hodin (např. SQL server vyžaduje jiné nastavení než WEB server). Pro jednotlivé části monitoringu pak existují předdefinované šablony, tzv. Knowledge Modules (KM) např.:

- PATROL KM for Microsoft Windows Operating System
- PATROL KM for Microsoft Windows Active Directory
- PATROL KM for Microsoft Windows Domain Services
- PATROL KM for Microsoft Cluster Server
- PATROL KM for Log Management

Veškeré výsledné nastavení pro tento jeden konkrétní server je uložen v konfiguračním souboru, který je uložen lokálně na každém serveru. To umožňuje na straně jedné značnou míru standardizace (např. při instalaci nového serveru se použije standardizované šablony), ale na straně druhé i značnou míru individuálního nastavení jednotlivých serverů podle jejich skutečného využití. Konkrétní stav takového nastavení na konkrétním serveru znázorňuje obrázek 2.



Obrázek 2: Vzhled konfiguračního stromu monitorovaných komponent PATROL Agenta

Shora dolů lze postupně vidět všechny monitorované komponenty od logických disků přes monitoring hardware až po monitoring jednotlivých jader procesoru a jejich monitorované parametry. Mezi jinými jde o parametr „CPUprcrProcessorTimePercent“. V pravé části strom dále pokračuje evidencí security informací přes systémové informace až po Windows logy (aplikační a systémový).

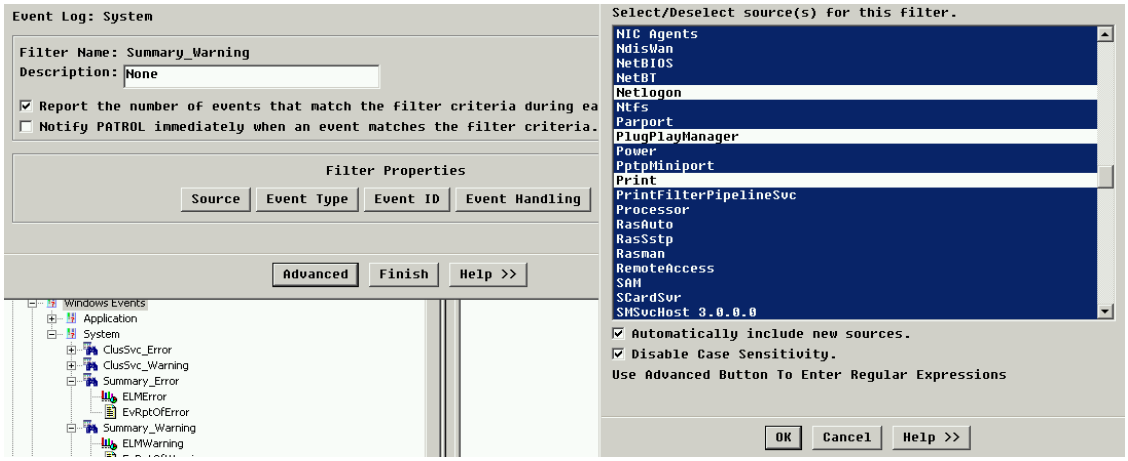
PATROL Agent pracuje relativně autonomně. Po načtení konfigurace z tzv. PATROL Knowledge Modules to znamená, že je schopen monitorovat, generovat upozornění, zaznamenávat historii stavů a management serveru provádět bez přímé konektivity na PATROL konzoli (tedy centrální management).

1.3 Způsob nastavování PATROL Agentů

PATROL Agent může být nastavován dle potřeby zcela nezávisle server od serveru. Jeho výchozí konfigurace je dána tzv. Baseline (základní, výchozí sadou nastavení pro všechny monitorované parametry). Baseline je oficiální dokument schválený na Windows technology meetingu senior specialisty a softwarovými architekty firmy. Jeho změna podléhá schvalovacímu řízení a žádná změna nesmí být provedena unáhleně na základě jednotlivého požadavku bez náležité analýzy jeho dopadu. V omezené míře mohou být povoleny částečné odlišnosti pro specifická prostředí různých zákazníků. I tyto změny však musejí být předem schváleny. Pokud však je změna schválena, je potřeba dohodnutou změnu provést. Typicky je pak potřeba provádět změny najednou na několika desítkách až stovkách serverů.

1.3.1 Manuální způsob nastavení PATROL Agentů

Konfigurační soubor PATROL Agenta, který je uložen přímo na serveru je možno konfigurovat jednotlivě pomocí „Agent Admin Console“ jako na obrázku 3, odkud máme přístup na všechny monitorované servery. Odsud lze pak měnit nastavení monitoringu pro každý jednotlivý server a upravovat každý monitorovaný parametr individuálně.



Obrázek 3: Konfigurační okna pro jednotlivé monitorované zdroje a jejich atributy

Nevýhody takovéto ručně prováděné změny nastavení jsou nasnadě. Monitorovaných instancí jsou na každém serveru typicky desítky a každá instance má další parametry, podle kterých je či není generován příslušný „alarm“. Možnost lidské chyby je značná.

Těmto chybám se lze vyhnout a zároveň dosáhnout značného zvýšení produktivity práce díky využití skriptování z prostředí produktů jako je BEM či BSA.

1.3.2 Centrální způsob nastavení PATROL Agentů

Dalším možným způsobem konfigurace PATROL Agenta je hromadné nastavování jeho parametrů. PATROL Configuration Manager podporuje centrální instalaci PATROL Agentů. Cílový stav je takový, aby bylo možno jedno a totéž nastavení aplikovat na jakýkoliv server kdykoliv bude potřeba bez ohledu na konkrétní prostředí serveru. To samozřejmě vyžaduje využití sofistikovanějšího přístupu, než procházení konfigurace bod po bodu. Firma BMC poskytuje obrovské množství nastavovacích pravidel „Rulesets“ pro zajištění defaultního (výchozího) nastavení pro jednotlivé „Knowledge Modules“.

Tyto „Rulesets“ jsou pak uloženy přímo na lokálním disku serveru, kde je PATROL Agent naistalován. Výpis 2 představuje krátký výtah konfiguračního skriptu pro nastavení monitoringu služby DHCP.

```
PWK__PKMforMSWinOS_config/ServiceMonitoring/ServiceList/Dhcp/Alarm" = { REPLACE = "1" },

"...MSWinOS_config/ServiceMonitoring/ServiceList/Dhcp/AutoRestart" = { REPLACE = "1" },
"...MSWinOS_config/ServiceMonitoring/ServiceList/Dhcp/IgnoreAutoResetConfig" = { REPLACE = "0" },
"...MSWinOS_config/ServiceMonitoring/ServiceList/Dhcp/Monitor" = { REPLACE = "1" },
"...MSWinOS_config/ServiceMonitoring/ServiceList/Dhcp/MonitorNotRespond" = { REPLACE = "0" },
"...MSWinOS_config/ServiceMonitoring/ServiceList/Dhcp/MonitorProcess" = { REPLACE = "0" },
"...MSWinOS_config/ServiceMonitoring/ServiceList/Dhcp/ServiceName" = { REPLACE = "Dhcp" },
"...MSWinOS_config/ServiceMonitoring/ServiceList/Dhcp/StartupType" = { REPLACE = "Automatic" },
"...MSWinOS_config/ServiceMonitoring/ServiceList/Dhcp/WarningAlarm" = { REPLACE = "0" },
```

Výpis 2: Výtah konfiguračního souboru pro nastavení PATROL Agenta

Stejným způsobem pokračuje nastavení všech monitorovaných tříd a parametrů. Jen pro nastavení monitoringu samotného operačního systému má tento konfigurační soubor běžně stovky a v některých případech i tisíce řádků.

2 AUTOMATIZACE MONITORINGU

Ve velkých firmách je monitoring součástí tzv. Event Managementu, tedy správy událostí, dle specifikace ITIL. Smyslem monitoringu je nejen stav IT infrastruktury monitorovat, ale také v maximální možné míře automatizovat. A to jak vlastní monitoring, tak reakce na nastalé stavy. Významným očekáváním všech ředitelů IT společností je automatizací uspořit pracovní náklady na správu IT infrastruktury. Nicméně, realita je mnohdy této skutečnosti značně vzdálena. Ne vždy se podaří automatizační nástroje nasadit optimálním způsobem tak, aby opravdu k úspoře pracovní síly došlo. Problémem je zejména vysoká míra rizika volby mezi 100% spolehlivostí požadovanou zákazníky a snahou o úsporu nákladů na dohled na straně poskytovatelů služeb. Pokud se půjde pouze cestou úspory nákladů bez investice do opravdu fungujícího programového vybavení, pak pouhá redukce množství monitorovaných událostí povede ke snížení možnosti predikce nebezpečných stavů a snížení kvality služeb. V konečném důsledku může dojít až ke ztrátě důvěry a pak i ke ztrátě zákazníka samotného.

Pokud je ovšem motivace k automatizaci vedena nejen úsporou pracovních sil, ale právě snahou o dosažení vyšší kvality služby, pak je dosažení cíle pravděpodobně reálnější. Je to právě oblast spolehlivosti, kde nasazení automatizace výrazně předčí rizika lidského faktoru. Tím jsou myšleny zejména chyby v úsudku a možnost výskytu opomenutí při vykonávání opakovaných úloh.

Automatizace je tedy významným faktorem ke zvýšení spolehlivosti při nastavování velkých počtů opakujících se úloh. Jelikož je však samotný proces automatizace velmi náročný na práci vysoce kvalifikovaných specialistů napříč všemi IT specializacemi (operační systémy, sítě, skriptování, programování), je automatizace efektivní právě jen při hromadném nasazení na stovkách až tisících instancích. Se zvyšujícím se počtem instancí jednotlivých úloh může být efekt automatizace markantní a v konečném důsledku může uspokojit i vedoucí pracovníky očekávající hlavně úsporu pracovních sil. Dle mého názoru by však motivace měla být postavena hlavně na požadavku zvýšení kvality.

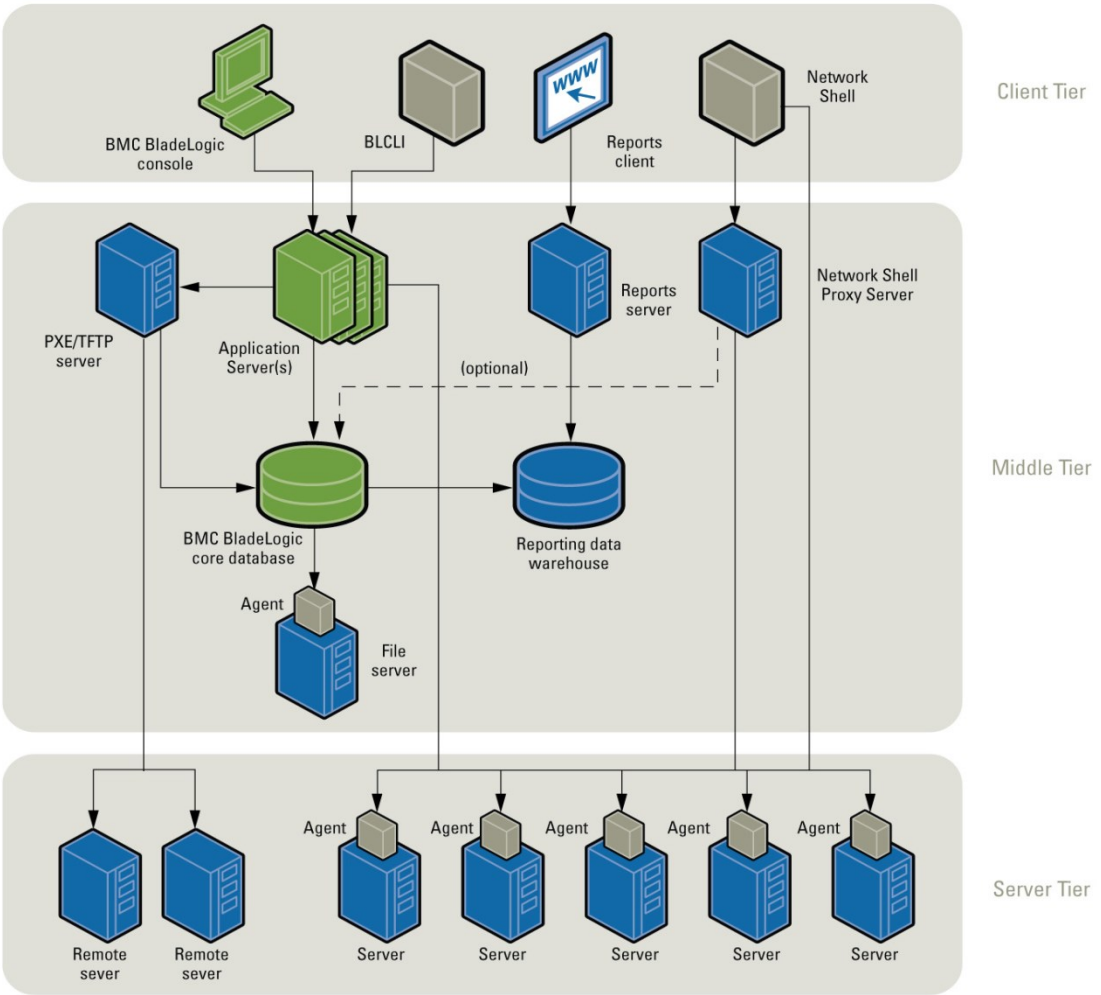
2.1 Prostředí produktu BMC Server Automation

BMC Server Automation (dále BSA) je serverový systém umožňující administraci a automatizaci jednotlivých činností a správu velkých IT systémů. Tento systém poskytuje relevantní informace o serverech pro řízení těchto prostředí, eviduje jejich změny, umožňuje inovace. To vše bez nutnosti znát značné množství detailů o spravovaných technologiích. Dále umožňuje automatizaci procesů jako je „patchování", tedy instalace opravných balíčků operačních systému a aplikací. Zjednodušuje evidenci konfigurace serverů a následný reporting aktuálního stavu nebo změn. Zajišťuje konzistenci, vyhodnocuje shodu a zpřístupňuje komplexní pohled na datová centra. Typicky automatizuje všechny opakující se činnosti. V ideálním případě umožňuje všechny tyto operace realizovat prostřednictvím jednoho agenta, který pracuje za operátora. Není tedy potřeba se individuálně přihlašovat na server, veškerá práva má již přidělena naistalovaný agent. Tímto způsobem je možné spravovat až tisíce serverů

z jednoho místa. BSA však není pouze o agentech. Bylo by hodně zjednodušené pohlížet na agenta pouze jako na prodlouženou ruku správce. Agent je pouze součást velmi komplexního nástroje pro správu a automatizaci.

2.2 Popis BSA infrastruktury

Jelikož je BSA opravdu velmi komplexní nástroj, pozornost bude zaměřena pouze na komponenty přímo související s potřebou spuštění automatizačních jobů na serverech. Základní představu o celém prostředí si lze udělat při pohledu na obrázek 4. Operátor má přehled o celé infrastruktuře prostřednictvím administrátorské konzoly (BMC BladeLogic console). Odtud vidí správce celou infrastrukturu, provádí skriptování, analýzu a spouští joby na jednotlivých serverech nebo skupinách serverů.



Obrázek 4: Orientační schéma infrastruktury BSA

Vlastní logiku obstarávají velmi výkonné aplikační servery (Aplications Server v Middle Tier vrstvě), kde je umístěna celá výkonná část včetně napojení na databáze a reportingového nástroje. Aplikační servery zajišťují zpracovávání jobů, linkování skriptů, předávání parametrů jednotlivým skriptům a jejich zaslání na příslušného agenta na serveru. Dále se starají o řazení jednotlivých požadavků do fronty, uchovávají výsledky jednotlivých jobů. Současně aktualizují obsah tzv. Smart Groups, tedy chytrých skupin serverů vytvořených na základě sady kritérií.

V neposlední řadě je zde RSCD agent. Tedy agent nainstalovaný přímo na jednotlivých serverech. Právě jeho prostřednictvím a jeho přístupovým právům (typicky práva server administratora) je zajištěna konektivita do prostředí operačního systému. Těmito oprávněními zajišťuje neustálou připravenost k jakékoliv operaci na serveru.

2.3 NSH skriptování

NSH je soubor příkazů pro změny vycházející z logiky a struktury jazyka UNIX Bash (Steve Shah, 2007). Rozdíl je, že pomocí NSH příkazů lze prostřednictvím RSCD agentů přistupovat jak k lokálním, tak vzdáleným souborům bez nutnosti mít práva k souborovému systému vzdáleného serveru. Taktéž není nutné vytvářet spojení na vzdálenou plochu serveru a ověřovat přístup za pomoci autentifikačního mechanismu pro vzdálené přihlašování. Není tedy nutné se přihlašovat na server pomocí Remote Desktop, či jiným způsobem.
Použitím NSH příkazů lze spravovat celou rozsáhlou IT infrastrukturu jako je datové centrum se servery Unix i Windows, jakoby se jednalo o jeden server. Lze spouštět administrativní funkce na velkém množství hostů z jednoho centrálního počítače namísto logování a používání funkcí jako je „Telnet" a přihlašování se na jednotlivé servery. To umožňuje mnohem rychleji provádět změny na spravovaných serverech. Jedná se například o změny v jednotlivých souborech na lokálních discích, jejich kopírování ze serveru nebo na server. Spuštění jednotlivých příkazů v příkazové řádce serveru, spuštění a zastavení služeb. To vše mnohem komfortněji, z jednoho místa, bez nutnosti tyto příkazy provádět lokálně v jednotlivých „command line" (příkazových řádcích).

2.4 Příklady NSH skriptů

Výpis 3 zobrazuje ukázku jednoduchého skriptu, který provede jednorázový úkon na cílovém serveru. Konkrétně spustí speciální příkaz jazyka NSH vracející „hostname" serveru. Vykonání

```
#!/bin/nsh

hostname=${NSH_RUNCMD_HOST}
echo "${hostname}"
```

Výpis 3: Ukázka NSH skriptu vracejícího hostname serveru

tohoto příkazu je nezávislé na platformě a příkaz bude vykonán v prostředí kteréhokoliv OS. V záhlaví skriptu je vidět identifikaci jazyka, v němž je kód napsán: „`#!/bin/nsh`“. Výpis 4 uvádí jednoduchý příklad vykonání cyklu. Vlastní příkaz je proveden vždy pouze na jednom konkrétním cílovém serveru.

```
#!/bin/nsh
array_of_hosts=$1

for server in ${array_of_hosts}; do
  echo "Server: ${server} has hostname: "
  nexec -n ${server} hostname
done
```

Výpis 4: Ukázka cyklu v jazyce NSH

Seznam serverů, na kterých se má tento skript postupně sériově vykonat je předán jobu jako parametr „%h“. Skript si převezme proměnnou „%h“ do pole „array_of_hosts“, z něhož je postupně předávána v cyklu „for“ proměnné „server“ jednotlivým skriptům. Tato volba je vhodná, zejména je li potřeba mít kontrolu nad provedením každého jednotlivého příkazu, který se za pomoci příkazu „echo“ loguje. Výpis 5 vysvětluje provedení příkazu na příkazové řádce cílového serveru. Skript spustí program cmd.exe (tedy příkazový řádek) a vrátí PID (identifikátor tohoto běžícího procesu).

```
#!/bin/nsh
array_of_hosts=$1

for server in ${array_of_hosts}; do
      Result1=`nexec -n ${server} cmd.exe /c "TASKLIST /FI
      "imagename eq cmd.exe" /svc"`
      echo "\n ${server}, ${Result1} " >>
      //serverName.domain.LOCAL/temp/chovalum/list.txt
done
```

Výpis 5: Provedení příkazu na příkazové řádce serveru s přesměrováním výstupu do souboru

Jelikož je skriptem zajištěno spuštění příkazového řádku, bude tento vždy aktivní a musí tak dojít k zachycení minimálně jednoho běžícího procesu s názvem cmd.exe. Výsledek provedení tohoto cyklu spuštěného na třech cílových serverech je patrný na výpisu 6. Tento výpis se nachází rovněž ve výsledcích jobu, který je pro tento účel vytvořen. Za pomoci jazyka NSH lze provádět veškeré příkazy, které jsou známy v jednotlivých prostředích bez ohledu na platformu. Tedy jak pro operační systém Windows (Annette Stolz, 2003), tak UNIX (Steve Shah, 2007). Jazyk NSH je nezávislým prostředníkem mezi těmito jinak zcela odlišnými světy. Kompletní seznam všech příkazů jazyka NSH je k dispozici v referenčním manuálu firmy BMC (BMC, 2011).

```
SERVER1.DOMAIN.LOCAL,
Image Name                     PID Services
=========================================
cmd.exe                       3272 N/A


SERVER2.DOMAIN.LOCAL,
Image Name                     PID Services
=========================================
cmd.exe                       2824 N/A


SERVER3.DOMAIN.LOCAL,
Image Name                     PID Services
=========================================
cmd.exe                        584 N/A
```

Výpis 6: Výpis záznamu provedení jobu pro zjištění PID běžícího procesu na serveru

2.5 Automatizace a bezpečnost

Při práci na serverech je potřeba mít se velmi na pozoru. Obecně prospěšné technologie, díky nimž se používají skripty a automatizace, mohou být totiž také využívány k vytváření škodlivého softwaru. Klíčovou schopností, jejíž zvládnutí je pro další práci nezbytné je dovednost nastavit automatizační prostředí tak, aby povolovalo spouštět žádané skripty pro správu a současně chránilo servery před škodlivými skripty (Doj Jones, 2006).

2.5.1 Metody zabezpečení

Základní princip zabezpečení u skriptů znamená zajistit spuštění pouze námi vytvořených skriptů a současně zamezit spuštění skriptů zavlečených a nežádoucích. Mezi způsoby jak zajistit rozeznání autorizovaného skriptu patří v současné době zejména digitální certifikace (digitální otisk).

Identifikace skriptu důvěryhodným certifikačním úřadem je snadná a nevyžaduje žádné zvláštní úsilí. Postačuje vlastnit dostatečně důvěryhodnou certifikační autoritu, případně si takovou identitu zajistit u komerčních poskytovatelů digitálních certifikátů. Princip certifikačních autorit je v současné době již dostatečně znám. Důvěryhodná certifikační autorita díky digitálnímu certifikátu, kterým se skript podepíše, ověří nejen jeho původ ale i integritu. To potvrdí, že obsah skriptu nebyl od jeho vytvoření neoprávněně změněn (Doj Jones, 2006).

Pokud se využije digitální certifikace v praxi, pak před spuštěním skriptu dojde nejprve k ověření, zda je skript podepsán důvěryhodnou certifikační autoritou. Ne každý certifikační úřad může být totiž považován automaticky za důvěryhodný. Windows přímo obsahuje seznam takovýchto nedůvěryhodných autorit a jimi podepsané skripty jsou tedy automaticky odmítnuty.

Při vzdáleném spouštění skriptů přichází do hry další faktor, a to je identita uživatelského nebo systémového účtu, pod jehož pověřením se skript spouští. Jak je známo, prostředí serverů má implementováno standardně poměrně rozsáhlou skupinu účtů poskytujících dostatečnou škálu privilegií nutných ke spouštění jak programů, tak skriptů. Jejich dostatečně důkladným návrhem lze ošetřit jak identifikaci bezpečného skriptu, tak problémy při nedostatečných právech pro provedení vlastních operací, jenž jsou součástí těla skriptů.

Do hry dále vstupují také i další ochranné prvky, a to brány „firewall“. I na tyto omezení je potřeba při návrhu a práci se skripty pamatovat. V prostředí velkých počítačových sítí se však jedná o obsáhlou problematiku, která zde nebude detailněji objasněna.

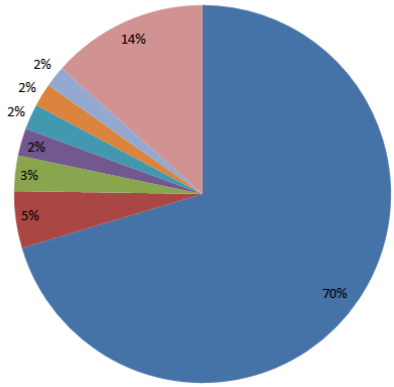
3 ROZBOR POŽADAVKŮ NA NASTAVENÍ MONITORINGU

Při výběru konkrétních úloh pro automatizaci nastavení monitoringu se vycházelo z reálného prostředí analýzou všech skutečně vygenerovaných událostí. Těchto tzv. Eventů nebo jinak řečeno Automatických Incidentů bylo v konkrétním prostředí firmy zachyceno měsíčně cca 300 000. Výběr událostí pro automatizaci nastavení monitoringu byl určen pomocí Paretova pravidla[1] analýzou všech událostí evidovaných za poslední rok.

V potaz byly rovněž vzaty požadavky specialistů a jejich zkušenosti s různou náročností nastavování. Shrnutí důvodů k výběru právě uvedených událostí a požadavky na jejich nastavení následuje v kapitole 3.1.

3.1 Analýza událostí CPU

Monitoring událostí CPU je velmi důležitý a vyžaduje vysokou prioritu. Žádná událost tohoto typu by neměla být zanedbána. Původně byly všechny události vždy předány k ruční kontrole. Počet takových událostí dosahoval řádově tisíce za měsíc. Z analýzy těchto automatických incidentů (obrázek 5) vyplynulo, že 70 % všech zaznamenaných událostí překročení monitorované úrovně vytížení procesoru je v okamžiku následné kontroly již stabilizováno. Od výskytu monitorované události do okamžiku jejího zániku uplyne doba maximálně tří hodin.

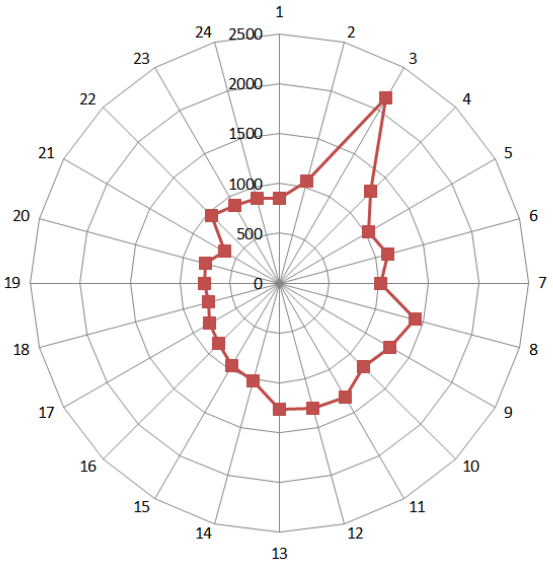


pořadí	Doba řešení v hodinách max.	počet	% z celkového počtu
1	3	17 982	70,4%
2	6	1 239	4,8%
3	9	778	3,0%
4	12	597	2,3%
5	15	556	2,2%
6	18	505	2,0%
7	21	461	1,8%
8	delší než 24	3 429	13,4%
Celkem		25 547	100%

Obrázek 5: Výsledek analýzy pomocí Paretova pravidla pro monitorované události přetížení CPU

[1]Vilfredo Federico Damaso Pareto (15. červenec 1848 - 19. srpen 1923), italský ekonom: Výsledek analýzy dle Paretova pravidla ukazuje kumulovaný součet výskytů jednotlivých příčin, podle jejich procentuálního zastoupení v souboru dat. Tyto jevy se pak seřadí od nejčetnějších po nejméně se vyskytující. Pak je možno například prohlásit, že buď 20 %, nebo 80 % (záleží na pozici pozorovatele) je možno zanedbat, neboť těchto prvních 20 % příčin reprezentuje 80 % všech událostí a 80 % příčin lze zanedbat, neboť reprezentují pouze 20 % událostí.

Výše uvedené znamená, že se jednalo o dočasné překročení vytížení procesoru a nebylo nutné se takto vygenerovanému alarmu vůbec věnovat. Respektive, než došlo ke kontrole, bylo vytížení CPU již ošetřeno jiným způsobem zahrnujícím například automatický restart kritické služby za pomoci jiných opravných mechanismů. U kritických služeb je monitoring zajištěn samostatnou cestou. Například u SQL serverů jsou monitorovány fronty čekajících jobů. Zajímavý je rovněž graf zachycující počet zaznamenaných událostí překročení vytížení CPU v závislosti na denní době (obrázek 6). Z grafu jednoznačně vyplývá, že největší problémy dělají automatické zálohovací joby naplánované obvykle v nočních hodinách (typicky mezi 22. hodinou a 4. hodinou ranní). Pak nastává pokles a opětovný nárůst v brzkých dopoledních hodinách (špička v 8 hodin) a viditelný pokles po skončení typické pracovní doby (15 hodin). Během dne nastává další pozvolný pokles. Z tohoto vyplývá, že události přetížení v nočních hodinách rovněž není potřeba nijak významně sledovat. Zálohovací joby jsou rovněž sledovány samostatnou monitorovací službou. Pokud taková záloha není dokončena do ranních hodin, je automaticky stornována. Specialisté zodpovědní za zálohování jsou tak informováni a mohou pracovat na nalezení příčin takového selhání.



Obrázek 6: Závislost počtu událostí překročení vytížení CPU na denní době

3.1.1 Příčiny přetěžování CPU

Procesor je jedna z klíčových komponent fungování jakéhokoliv počítače či serveru. Z tohoto důvodu je považován monitoring CPU za kritický a je nezbytně nutné hodnotu vytížení CPU sledovat a vyhodnocovat. Na straně jedné je tu logický požadavek na smysluplnou investici a tedy potřeby zakoupení právě takového procesoru, který odpovídá reálným potřebám požadovaného výkonu.

Na straně druhé by nemělo docházet k přetěžování využití procesoru. Smyslem správně nastaveného chodu serveru je zajistit využití procesorového času tak, aby měly všechny aplikace běžící na daném serveru přístup k času CPU kdykoliv si o něj požádají. Mezi nejběžnější příčiny přetížení procesoru (za předpokladu, že systém je optimálně navržen a nedochází k přetížení za předpokládaného provozu) patří:

- neočekávané „zatuhnutí“ služeb systému
- neočekávané „zatuhnutí“ kterékoliv ze spuštěných aplikací
- kolize spuštěných aplikací (nekompatibilita verzí)
- neočekávané provozní vytížení (nesprávný design provozovaného systému)
- virový útok
- nedostatek virtuální paměti
- spuštění zálohy systému (v plánovaném čase)

3.1.2 Analýza variant nastavení hladiny pro monitoring

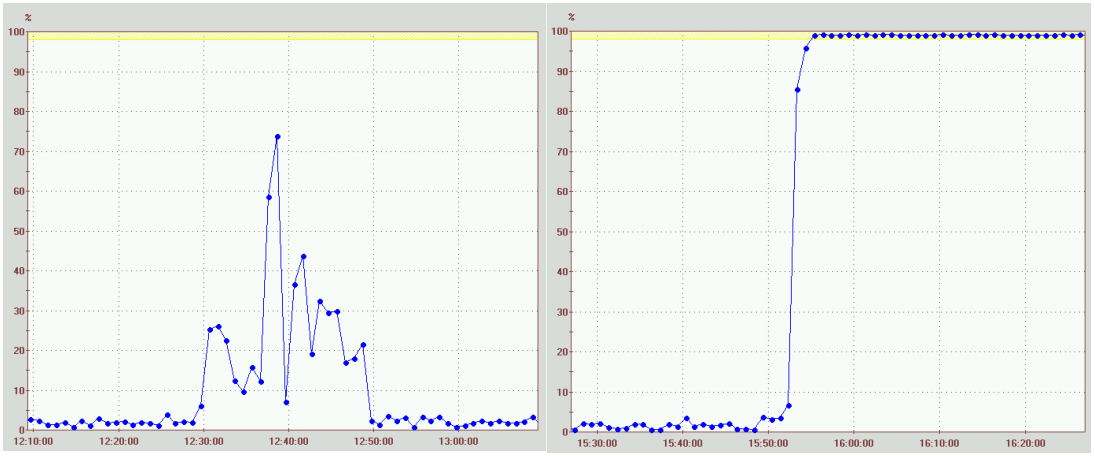
Z uvedených příkladů přetížení CPU a z provedené Root Cause analýzy je zřejmé, že podstatou monitoringu není zajímat se o stavy, kdy je procesor vytížen pouze například na 10 %. Informace o takovém využití CPU nemá pro zajištění bezpečného chodu aplikací žádnou relevantní hodnotu. Není jej tedy potřeba sledovat ani nijak o tomto využití CPU být informováni, a to ani z preventivních důvodů. Pro vlastní monitorování vytížení procesoru používá PATROL Agent třídu „CPUprcrProcessorTimePercent“. Parametry této třídy s doporučenými hranicemi pro vygenerování chybové zprávy jsou v tabulce 1.

CPUprcrProcessorTimePercent	BORDER	ALARM	0 - 100	Inactive
	ALARM1	WARN	90 - 95	Inactive
	ALARM2	ALARM	95 - 100	After 35 minutes

Tabulka 1: Původní hodnoty nastavení úrovní monitoringu CPU

Pro status „Warning“ není potřeba nastavovat hodnotu monitoringu. Monitoring hladiny „Alarm1-Warming“ je vypnut (Inactive). Nastává otázka, jakou hranici využití procesoru lze považovat za kritickou a jakou ještě ne. Teoreticky jsou všechny stavy využití pod 100 % nezajímavé a zajímavý je tedy pouze stav, který systém označuje jako využití na 100 %. Při dosažení této maximální možné hladiny je zřejmé, že již nelze vyřizovat požadavky na procesorový čas a v reálném čase a dochází ke zpomalování běhu aplikací. Procesor je v této situaci plně zaměstnán vyřizováním fronty požadavků aplikací na přidělení své taktovací frekvence. Všechny „timesloty“ jsou obsazeny a aplikace musí na volný „timeslot“ čekat. V reálném čase dochází ke zpomalení chodu aplikací, přičemž míra zpomalení záleží na délce fronty čekajících aplikací. Pro hladinu „Alarm2-Error“ je nutné parametr pro sledování využití CPU zapnout. Hladina, při které dojde k vygenerování této chyby, se nastaví na hodnotu blízkou 100 %. Jakou konkrétně zvolit výši této hodnoty, je otázkou k diskusi. Je teoreticky lhostejno,

zda to bude 98 % nebo 99,99 %. Z praxe lze vysledovat, že průběh křivek vytížení CPU má dvě základní charakteristiky. Buď se jedná o změny, kdy v rámci delšího časového období dochází jen k drobným výkyvům mimo typickou provozní hladinu (obrázek 7 vlevo) a nebo se jedná o prudký nárůst vytížení, který skokově dosáhne hladiny 100 % a v ní po nějakou dobu setrvává (obrázek 7 vpravo).



Obrázek 7: Typické varianty průběhu vytížení CPU

Pro objektivní posouzení správnosti nastavení se provedla analýza historicky dostupných událostí vygenerovaných v rámci dosavadního nastavení monitoringu. Na základě těchto uvedených předpokladů bylo rozhodnuto, že bude prodloužen interval, kdy se bude čekat na vygenerování chybové zprávy na dobu minimálně 120 minut. Po této době se již dá předpokládat, že se jedná o trvalé přetížení, které může způsobovat reálné problémy. Zároveň byla posunuta hranice, od které bude přetížení procesoru považováno za reálný problém na 98 %. V tabulce 2 je zobrazen výsledný požadavek na nové nastavení monitoringu CPU.

CPUprcrProcessorTimePercent	BORDER	ALARM	0 - 100	Inactive
	ALARM1	WARN	90 - 98	Inactive
	ALARM2	ALARM	98 - 100	After 120 minutes

Tabulka 2: Nové hodnoty nastavení úrovní monitoringu CPU

Byť se jedná o požadavek na změnu pouhých dvou parametrů, jejich ruční změna na jednom serveru s 16 jádrovým procesorem zabere spoustu času. A navíc se lze během této ruční konfigurace dopustit chyb. Jedná se tedy rozhodně o příkladného kandidáta na automatickou konfiguraci.

3.2 Analýza událostí NTFS

Mezi dalšími kandidáty na provedení změny v nastavení byl vybrán soubor událostí zaznamenaných v systémovém logu operačního systému Windows, tedy WINDOWS SYSLOG. Stejným způsobem (analýzou dle Paretova pravidla) bylo zjištěno, že 80 % všech událostí náleží pouhým šesti skupinám událostí, viz tabulka 3. Na prvním místě je zdroj událostí Microsoft-Windows-GroupPolicy. Jako další, s ohledem na zkušenosti specialistů, byl vybrán monitoring diskového subsystému, konkrétně zdroj událostí NTFS. Události z „Microsoft-Windows-FailoverClustering“ a „NICAgents“ jsou příliš komplexní problémy, které ve většině případů nelze snadno řešit pouhým přenastavením monitoringu.

Název sledovaného parametru	Procento výskytu sledovaných událostí
Microsoft-Windows-GroupPolicy	**22,00%**
Microsoft-Windows-FailoverClustering	15,05%
NICAgents	14,77%
Ntfs	**14,01%**
Disk	6,69%
MonitoringTest	5,74%

Tabulka 3: Výstup analýzy podle Paretova pravidla pro události z logu Windows Syslog

Výběr monitoringu NTFS pro automatickou změnu nastavení určily dva základní argumenty:

a) Jedná se o známou chybu, která se vyskytuje pouze na serverech s novou komponentou pro zálohování Wbadmin. Ta při vytváření zálohy na systémový disk provádí tuto zálohu za pomoci virtuálních VHD disků (Microsoft, 2013). Tuto chybu lze v tomto případě ignorovat.
b) Provedení změny nastavení tohoto parametru je velmi náročná. Vyžaduje velmi přesný způsob provedení filtru takové události a v případě chyby v provedení nastavení hrozí odfiltrování i velmi kritických chyb systému NTFS.

3.2.1 Požadavky na nastavení monitoringu

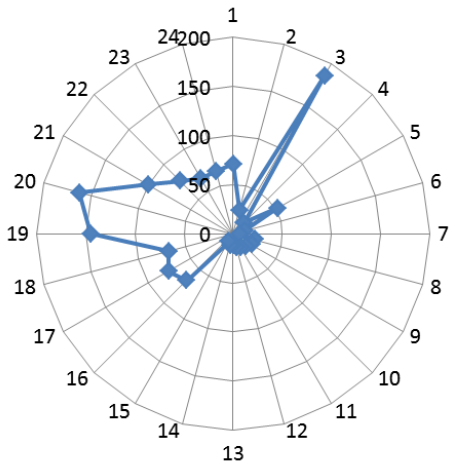
Smyslem monitoringu diskového subsystému NTFS, jenž je v současnosti standardem na serverech s operačním systémem Windows, je zajistit bezpečnost přístupnost dat pro každý požadavek systému nebo aplikace. U chybových zpráv ze zdroje NTFS bylo dalším rozborem (Microsoft, 2013) zjištěno, že se jedná o problémy související s chybovými kódy dle tabulky 4.

Error ID	Popis chyby	Zastoupení
137	The default transaction resource manager on volume …	33,89%
55	The file system structure on disk is corrupt and unusable	27,29%
57	The system failed to flush data to the transaction log	10,01%
141	{Delayed Write Failed} Windows was unable to save all the data for the file…	9,86%

Tabulka 4: Výstup analýzy dle Paretova pravidla pro chybové kódy NTFS

Root Cause analýzou těchto událostí byly zjištěny tři možné scénáře výskytu chyby:

a) Chyby 137 a 55 spolu souvisejí a jejich nejčastější příčinou je spuštění Windows backup (Wbadmin), který pro svou práci vytváří dočasný VHD disk, který však nelze žádným způsobem kontrolovat a jenž standardně po ukončení procesu zálohování má být ze systému odstraněn. Problém se vyskytuje pouze na serverech se systémem WINDOWS 2008 a vyšším, protože až od těchto verzí OS je využíván k zálohování systémových souborů Wbadmin. Bohužel v některých případech nedojde ke korektnímu odebrání tohoto virtuálního logického disku ze systému. Jedná se tedy o typické chování tohoto programu a nelze jej predikovat ani mu zabránit. Microsoft tuto chybu systému dosud neodstranil. Časová analýza na obrázku 8 dokazuje časovou závislost s časy, kdy se provádí zálohování pomocí komponenty Wbadmin (večerní a noční hodiny).



Obrázek 8: Závislost počtu událostí NTFS na denní době

b) Stejné chyby mohou být však zároveň generovány i ve skutečně závažných případech poškození fyzického disku, například v případě poškození pevného disku a při skutečných chybách zápisu na disk. Všechny případy výskytu těchto chyb tedy musí být jednotlivě přezkoumány. Jak však odlišit případy, kdy se jedná o souvislost se spuštěním zálohy a kdy se může jednat o závažný problém? Analýzou situace bylo nalezeno jedno vodítko a tím je dodatečná informace obsažená v detailním popisu této události. Zde je však problém, neboť PATROL Agent nezasílá tak detailní informace z logu. Je tedy potřeba přímo nastavit PATROL Agenta na každém jednotlivém serveru tak, aby přímo PATROL Agent po přečtení detailních informací z logu rozlišil, o jakou zprávu se jedná. Rozlišovacím znakem je přítomnost informace, že se vygenerovaná chyba týká VHD disku. Detailní informace v logu tedy musí obsahovat variantně (podle verze operačního systému) tyto údaje:

- Description: The default transaction resource manager on volume VHD encountered
- Description: The default transaction resource manager on volume \\?\Volume{...

c) Ve třetím případě se jedná o známou chybu vyskytující se na virtuálních strojích, kde je disk počítače simulován. Firma VMware vydala následující Knowledge Base č.: 2006849 (Vmware, 2014), kde popisuje důvody výskytu těchto chyb u operačních systémů Window 2008 R2 instalovaných ve virtuálním prostředí. V případě, že je systém hostován na virtualizačních platformách společnosti VMware verze ESXi/ESX 4.1 nebo ESXi 5, je možno ignorovat celou skupinu chyb s chybovými kódy 50, 57, 137, 140 a 12289.

3.2.2 Varianty nastavení filtrování nepotřebných zpráv

Vyjmutí událostí souvisejících s VHD disky z monitoringu lze samozřejmě provést jako obvykle ručně. Zde je to však nesmírně komplikované a vyžaduje to sled mnoha kroků (kompletní manuál pro toto nastavení má osm stran), kde lze snadno udělat chybu. Může dojít dokonce k chybě kritické, k odstranění monitoringu všech chyb ze zdroje NTFS a tím se nedozvědět o pádu diskového subsystému serveru.

V rámci automatizace jsou tři možnosti:

- Provést nastavení agenta automatickým skriptem na všech serverech.
- Vytvořit kontrolní job, který spustí speciální kontrolní proceduru při každém výskytu této události. V případě že job vrátí pozitivní hodnotu (VHD disk byl nalezen), incident lze uzavřít jako nedůležitý.
- Vytvořit kontrolní job, který otestuje, zda se jedná o systém hostovaný na virtuální platformě společnosti Vmware. Pokud ano, lze incident uzavřít.

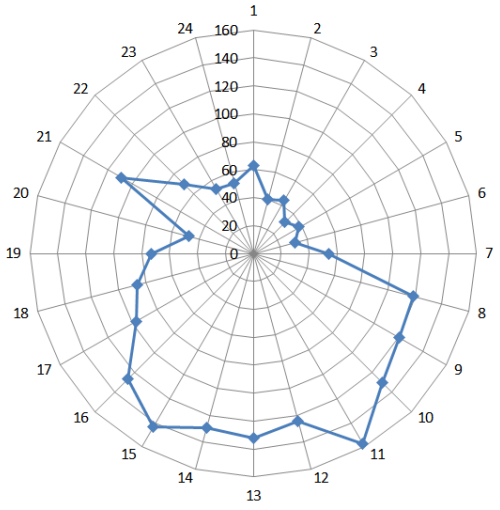
Jakou variantu zvolit je samo o sobě náročným úkolem. Roli pro vyhodnocení hrají tyto postoje:

- Zhodnocení náročnosti a spolehlivosti jobů
- Automatizace nastavení je náročnější než automatizace kontroly logu, ale jeho provedení je jednorázové. Po provedení nastavení není potřeba mít job dále aktivní. V případě změny podmínek je potřeba vytvořit nový konfigurační skript.
- Job pro kontrolu logu je relativně jednoduchý, ale vyžaduje soustavnou péči, ta však lépe reaguje na změnu podmínek.

Výsledek bussines analýzy není v době uzavření této bakalářské práce znám. Pro nastavení PATROL Agenta bude použit stejný nastavovací skript, jako v případě CPU, pouze se změní nastavované parametry. V případě doprogramování kontrolního jobu spouštěného na základě výskytu události NTFS mohou paralelně fungovat obě varianty. Programování automaticky spouštěných jobů na základě událostí je však nad rámec této bakalářské práce.

3.3 Analýza událostí Windows Group Policy

Jak je uvedeno v tabulce 3 (v kapitole 3.2) největší počet vygenerovaných událostí ze systémového logu Windows představují právě události související s funkcí Windows Group Policy. U této chyby je zřejmé, že množství událostí je vyšší v dobách, kdy jsou servery nebo pracovní stanice pravděpodobně využívány. Tedy v průběhu běžné pracovní doby, kdy se do sítě přihlašují uživatelské počítače a jsou vysílány požadavky na provedení nastavení GPO objektů, případně se provádějí replikace mezi doménami.



Obrázek 9: Časová analýza výskytu chyby GP

Časová analýza na obrázku 9 dokazuje závislost výskytu chyby na dobu obvyklé pracovní doby. Windows Group Policy je nástroj pro hromadnou správu oprávnění a nastavení aplikovaných na celý počítač a nebo na profil přihlášeného uživatele. Ve skupinách zásad je možné vytvářet kolekce nastavení, kterým se říká Group Policy Object (GPO). Ty dokáží měnit konkrétní parametry chování počítače nebo uživatele. Samotné nastavení GPO se pak aplikuje na jednotlivé Organizační jednotky (OU) v Active Directory (AD), čímž se zajistí aplikování nastavení jen na vybrané počítače nebo uživatele. Tímto způsobem lze spravovat potenciálně tisíce počítačů pouhou změnou jedné politiky GPO.

Stručný výčet toho, co všechno lze pomocí zásad skupiny implementovat:

- aplikování firemních standardů (skrytí ovládacích panelů, síťové tiskárny, spouštění skriptů)
- aplikování zabezpečení (změna oprávnění na určitých složkách, složitost hesla, skupiny s možností se lokálně přihlásit)
- hromadná instalace aplikací (Office, Adobe Reader, atd.)

3.3.1 Požadavky na monitoring událostí

Group Policy Objekty slouží k předávání nastavení registrů počítače (typicky uživatel připojující se do firemní domény). K předání GPO dochází vždy, když se uživatel přihlásí do domény. Vlastní nastavení je uloženo v souborech ve formátu ADMX (od verze Windows Vista a Windows Server 2008). Zásady jsou popsány pomocí standardů XML. ADMX soubory jsou uloženy v složce „%systemroot%\PolicyDefinitions“ a při vytváření nové politiky se nakopírují do adresáře SYSVOL na systémovém disku. Aplikace těchto ADMX souborů se prování jejich přenosem z doménového řadiče na lokální počítač. K tomuto přenosu dochází po přihlášení uživatele a pak každých 90 minut (interval může být na požadavek administrátora domény změněn). V případě výpadku přenosu je tato informace zalogována. Ojedinělý výpadek není většinou nijak závažná událost. Pokud ale nedojde k aplikaci GPO po delší dobu, ztrácí administrátor domény kontrolu nad počítači uživatelů a ti mohou mít problémy např. s funkcí některých aplikací vyžadujících speciální nastavení prostředí lokálního počítače. Proto je nutné rozlišit, kdy se jedná o problém dočasný a kdy o problém trvalý.

3.3.2 Varianty monitoringu

Z analýzy logů vyplývá (příloha č. 3), že v podstatě 100 % všech chyb generovaných pod

```
The processing of Group Policy failed. Windows could not authenticate to the Active Directory service on a domain controller. (LDAP Bind function call failed). Look in the details tab for error code and description.150120123349Invalid Credentials
```

Výpis 7: Výpis chybové zprávy zaznamenávající chybu 49 - Invalid Credentials

hlavičkou Group Policy má chybový kód 1006. Popis chybové zprávy je uveden na výpisu 7.

Root Cause analýzou bylo zjištěno, že lze ignorovat události s kódem „49Invalid Credentials“. Tyto události souvisí s problémem, kdy uživatelé jsou přihlášeni na svých počítačích a v průběhu tohoto přihlášení dojde k vypršení platnosti jejich hesla (Microsoft, 2013). Při následném intervalu pravidelného updatu GPO (v intervalu 90 minut) dojde k vygenerování této události. Na tuto událost není nutno nijak reagovat, neboť uživatel může i nadále pokračovat v práci a při následném přihlášení a odhlášení do domény je automaticky přímo systémem vyzván ke změně hesla.

I zde (stejně jako u NTFS) je možnost nastavit přímo PATROL Agenta ručně nebo pomocí konfiguračního skriptu a nebo zvolit možnost kontroly událostí pomocí jobu spouštěného touto událostí.

4 SKRIPTY PRO ZMĚNU NASTAVENÍ MONITORINGU

Jak bylo popsáno v teoretické části, pro vlastní automatizace je potřeba vždy nejprve vytvořit skript. Smyslem skriptu je provést konkrétní činnost na konkrétním serveru, který je konkretizován jako cílový až při spuštění v těle jobu. Job, který tento skript zavolá, předá skriptu konkrétní hodnoty pro jeho práci. Typicky se jedná o cílový server a případně další parametry. Ty lze specifikovat libovolně a dolaďovat konkrétní parametry pro vykonatelný kód skriptu. Skript pak provede vlastní úkony na tomto konkrétním serveru a vrátí návratovou hodnotu, pokud je vyžadována.

Samotný skript se programuje tak, aby jeho kód byl vykonatelný na jednom jediném konkrétním serveru. Protože se neví na jakém, je jazyk NSH vytvořen právě tak, aby byl kód vykonatelný vždy na jakémkoliv serveru. I přesto je však nutno na tuto univerzálnost stále pamatovat. Operačních systémů, byť i jen jednoho jediného výrobce je tolik, že nelze očekávat vykonání jednoho konkrétního příkazu vždy a bez jakýchkoliv komplikací. Základní složitost návrhu těla skriptu není ve složitosti očekávané akce, ale v nutnosti požadavku na univerzálnost. I když jazyk NSH tuto situaci značně zjednodušuje, v praxi se osvědčila metoda oddělování a psaní skriptů pro samostatná prostředí. Je zřejmé, že nejtypičtější oddělení bude mezi UNIX a WINDOWS. Nicméně v rámci WINDOWS je vhodné oddělovat verze, které jsou vývojově významně odlišné. Například operační systémy do verze Windows 2008 (včetně) vyžadují jiný přístup než verze 2008 R2 a novější. V praxi je pak vytvořen jen kód pro verze novější, neboť starších verzí operačních systémů postupně přirozeně ubývá. Starší verze pak navíc neumožňují využívání sofistikovanějších metod komunikace s prostředím serveru, například skriptovací jazyk Power Shell společnosti Windows.

Způsob automatizace, vytváření skupin serverů a psaní jobů je popsán v kapitole 5.

4.1 Metoda vývoje skriptu pro konfiguraci monitoringu

Jak je uvedeno v kapitole 1.3.2, nastavení PATROL Agenta lze automatizovat tím způsobem, že se přepíše konfigurační soubor. Ten PATROL Agentovi říká, které instance má agent monitorovat a jaké jsou podmínky pro vytvoření chybové zprávy. Náplní této práce není zkoumat tvar těchto konfiguračních zpráv. Analýzou požadavků na změnu nastavení byl učiněn závěr, že nelze přepisovat původní nastavení novým jako celek. Jak bylo uvedeno, konfigurační skript pro nastavení monitorovaných parametrů má celkem stovky až tisíce řádků. Jeho výsledný tvar byl dán především původním stavem při instalaci serveru, kdy bylo vytvořeno defaultní (základní) nastavení. Toto nastavení bylo následně několikrát změněno na základě požadavků zákazníka, nebo – častěji – bylo odladěno specialisty na základě Root Cause analýzy a změny z důvodu ladění systému během provozu. Aby bylo možné realizovat automatizaci nastavení monitoringu, je nutné změnit pouze nastavení monitoringu jedné monitorované instance (například CPU) na každém konkrétním serveru, což reprezentuje jen několik málo řádků konfiguračního souboru.

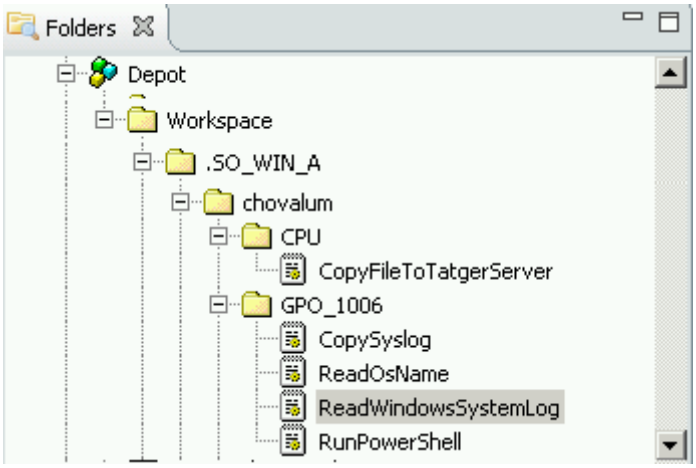
Požadavek na skript tedy je, aby uměl načíst původní konfiguraci, rozdělit ji, najít původní nastavení pro instanci, jejíž monitoring požadujeme změnit, vyjmout ji a nahradit novou konfigurací. Tu pak opět uložit na server a restartovat PATROL Agenta. Poté je nové nastavení načteno a celá práce s nastavením monitoringu pro danou instanci je hotova.

Vlastní vývoj skriptu byl prováděn spirálovou metodou, po analýze, následoval vývoj kódu, jeho testování a následná finální podoba byla uvolněna k použití.

Vyvinutý skript je vytvořen jako zcela univerzální, umožňuje tedy nahradit jakoukoliv část konfiguračního souboru PATROL Agenta jinou. Přestože vlastní skript je opravdu univerzální, není mi známo, pro jaké konkrétní nastavení bude v konečném důsledku použit mimo platformu Windows. Procesní řízení uvnitř firmy je totiž nastaveno tak, že neumožňuje efektivní koordinaci mezi těmito rozdílnými světy. Výjimkou je nastavení monitoringu CPU, kde byla použita obdobná analytická data. Výsledky změny nastavení monitoringu CPU jsou uvedeny souhrnně pro obě platformy.

4.2 Vytvoření skriptu pro změnu nastavení monitoringu

Aby bylo možno provádět automatickou změnu nastavení monitoringu, musí se nejprve vytvořit NSH skript. V prostředí BSA je nutno nejprve umístit vlastní skript na file server, kde bude kdykoliv k dispozici. Skript je vždy uložen ve složce s názvem Depot, která je dostupná z aplikačního serveru (jeho umístění v infrastruktuře BSA je patrné na obrázku 4 v kap. 2.2). Zde je operátorovi podle jeho aktuálních práv umožněn přístup na jeho vlastní složku, kde si může vyrobit svou vlastní strukturu složek dle svých zvyklostí tak, jak mu to vyhovuje. Na obrázku 10 je patrné umístění mé pracovní složky (Workspace) a její obsah.



Obrázek 10: Umístění skriptu a struktura složek v části Depot

Na obrázku 11 je zobrazeno okno s volbami provádění vlastního NSH skriptu. Tento konkrétní skript má zvolenu možnost „Execute the script once, passing the host list as a parametr to the script“. Skript bude spouštěn postupně na jednotlivých serverech předávaných jako parametr „%h“. Skript může obsahovat jakékoliv množství vstupních i výstupních parametrů (v záložce „parameters“) dle potřeby. Tento skript byl použit ve fázi ladění a umožňuje lepší kontrolu nad procesem ladění, neboť výstup tohoto skriptu byl logován do jednoho souboru sekvenční metodou a umožňoval tak kontrolu provádění skriptu nad cílovou skupinou serverů.

Obrázek 11: Volby nastavení NSH skriptu

Skript pro provedení změny nastavení monitoringu využívá volbu „Execute the script separatelly againtst each host“. Tato volba umožňuje souběžné spuštění na všechny servery a to opakovaně, tak dlouho dokud nebude na všech serverech jeho provedení úspěšné. Kód skriptu, je uložen pod záložnou „Script“. Tento kód nemůže být z důvodu ochrany autorských práv zaměstnavatele součástí tohoto textu.

4.3 Skript pro nastavení monitoringu CPU

Z důvodů velkého počtu provozovaných serverů a několika variant nainstalovaných PATROL Agentů bylo nutné vytvořit několik variant konfiguračních souborů. Obsah těchto konfiguračních zpráv PATROL Agenta není předmětem této práce a podléhá autorským právům v kompetenci oddělení správy monitoringu. Nicméně je známo, že konfigurační zprávy bylo nutno rozdělit na skupiny serverů, podle konkrétní verze PATROL Agenta. To proto, aby bylo aplikované nastavení kompatibilní s danou verzí PATROL Agenta a nedošlo k havárii monitoringu, což je považováno za kritický problém. Samotný NSH skript byl použit stejný. Princip, jak bylo dosaženo aplikace různých konfiguračních zpráv na skupiny serverů, je popsán v kapitole tvorby jobů.

Ověření nového nastavení pro monitoring CPU bylo po řádném otestování na testovacích serverech provedena na skupině firemních serverů mimo tzv. „produkční prostředí“. Tedy na serverech v majetku společnosti, nikoliv na serverech zákazníků. Tato validace proběhla v několika krocích během roku 2013 (klesající křivka na obrázku 17 v kapitole 6.1). Cílem byla vždy malá skupina maximálně sto serverů. Smyslem bylo zejména ověřit, zda nedojde ke kolizi

nastavení konfiguračního souboru PATROL Agenta. Právě při této validaci se odhalila některá úskalí související s nesourodostí konfiguračního souboru samotného PATROL Agenta, který jde na vrub jeho výrobce, tedy společnosti BMC. V případě ručního nastavení parametrů monitoringu na jednotlivých serverech totiž vzniká značná variabilita, která pak způsobuje nesourodost konfiguračních souborů. Tu již nelze zpětně odstranit. Konfigurační soubor musel nakonec zahrnout celou skupinu všech parametrů v jedné celistvé sekci. V případě CPU se jedná o všechna jádra procesoru a všechny jeho existující instance. Kromě instancí jednotlivých jader totiž někteří specialisté vytvořili i instance CPU Total, které monitorují součet vytížení všech jader na jednom serveru. U malého procenta serverů nebylo možno konfigurační soubor přepsat, neboť nebyl nalezen očekávaný řetězec. Nastavení takových serverů bylo nutno ponechat v jejich původních hodnotách. Takové servery zůstaly tímto skriptem nezasaženy a musely se dokonfigurovat ručně.

Po této prvotní validaci bylo provedeno ostré nasazení v plném provozu. Během tohoto procesu se již nevyskytovaly žádné neočekávané problémy a skripty byly dokončovány dle časového plánu. Z bezpečnostních a preventivních důvodů nebylo původně součástí NSH skriptu provedení restartu PATROL Agenta. Tímto byla vyloučena možnost hromadného výpadku v případě chybného nastavení. Vlastní restart PATROL Agenta byl ponechán až na okolnosti běžného provozu, kdy k restartu agenta docházelo nejčastěji v průběhu plánovaných odstávek v rámci např. pravidelného „patchování“ serverů, tedy instalace bezpečnostních balíčků společnosti Microsoft.

Průběh zde popisovaných prací se týká pouze WINDOWS serverů. Souběžně byly provedeny i změny nastavení monitoringu CPU pro UNIX servery, avšak hodnoty, průběh vlastních prací, problémy s nastavováním parametrů monitoringu CPU pro UNIX servery mi není znám. Výsledky jsou prezentovány souhrnně na datech za veškerou serverovou infrastrukturu.

4.4 Skript pro nastavení monitoringu NTFS

Požadavky na změnu monitoringu NTFS jsou náročnější na analýzu konfiguračního souboru PATROL Agenta. Jak bylo zmíněno, ruční konfigurace vyžaduje několik kroků a nastavení několika úrovní definujících přesná kritéria pro vyhodnocení chybové zprávy ze serveru. Po jeho identifikaci a nalezení je ovšem k vlastní změně nastavení opět využit univerzální skript pro přepis konfiguračního souboru. Aplikace tohoto nastavení se týká pouze omezené skupiny serverů, na kterých je využíván zálohovací program Wbadmin a nebo se jedná o virtuální stroje (dle provedené Root Cause analýzy, viz kapitola 3.2.1). Ve dvou případech chyb NTFS (viz tabulka 4 v kapitole 3.2.1) se jedná o známé chyby (Microsoft, 2013), které lze na základě jednoduchých kritérií ignorovat.

Pro implementaci nového nastavení monitoringu je využit modifikovaný skript popsaný v kapitole 4.1. Jeho úkolem je zkontrolovat přítomnost konfigurace pro monitoring parametru NTFS a tuto část nahradit rozšířenou specifikací pro kontrolu informace z logu, zda se jedná o chybu související s vytvořeným VHD diskem. Pokud ano, nebude PATROL Agent vytvářet na základě výskytu této události chybovou zprávu a informaci zahodí. Pokud však nebude

v systémovém logu operačního systému zachycena informace o virtuálním disku VHD, ale bude se jednat o reálný problém např. fyzického disku C, bude chybová zpráva vygenerována a předána k řešení.

Pro další možnosti týkající se identifikace virtuálního stroje nelze využít možnost konfigurace PATROL Agenta. Pro tyto případy byl vytvořen automatizační task. Ten je vyvolán v případě, že BEM systému bude doručena chybová zpráva ze serveru. Následně je vyvolán skript uvedený na výpisu 8. Ten testuje, zda se jedná o virtualizovaný operační systém. Pokud ano, může být událost uzavřena jako nepotřebná. Oba zde popsané způsoby, tedy nastavení PATROL Agenta a automatizační task, byly realizovány souběžně a vzájemně se doplňují. Nastavení monitoringu snižuje výskyt událostí souvisejících s VHD disky a automatizační task řeší situace vznikající na virtualizovaném operačním systému.

```
#!/bin/nsh

SERVER=${NSH_RUNCMD_HOST}
EXIT_CODE=1

SYSTEM_I=`nexec -e cmd /c systeminfo |grep "System Manufacturer"`
> /dev/null

STR_I=`echo $SYSTEM_I | awk '{print substr($0,22,6)}'` >
/dev/null

 if [ "$STR_I" = "VMware" ]; then
      EXIT_CODE=0
 fi

 exit ${EXIT_CODE}
```

Výpis 8: NSH skript pro identifikaci virtualizovaného operačního systému

4.5 Skript pro nastavení monitoringu Group Policy

Požadavek na změnu konfigurace monitoringu pro událost Group Policy s chybových kódem 1006 je prostý. Při výskytu řetězce s dodatečným popisem chyby obsahující kód „49Invalid Credentials“ lze tuto chybu rovněž ignorovat. Tuto skutečnost je PATROL Agent schopen identifikovat přímo na serveru a tím přímo zabránit vzniku chybové zprávy zasílané k další analýze. Na základě zkušeností s předchozími případy užití skriptu pro nastavení konfigurace, postačilo nyní nalézt správný konfigurační řetězec a ten aplikovat na skupinu serverů, kde je monitoring Group Policy aktivní. Tato aplikace je rovněž zajištěna prostřednictvím jobů popsaných v kapitole 5.

5 AUTOMATIZACE POMOCÍ ÚLOH (JOBS)

Architektura BMC Server Automation je navržena tak, že přímo vybízí k hromadnému nasazení a podřizuje tomuto cíli celou logiku svého prostředí. Nemělo by příliš smysl vyrobit skript a pak jej ručně spouštět na jednotlivé servery jeden po druhém. Aby byla usnadněna aplikace a umožněno řízení chodu skriptu, bylo vyvinuto speciální řešení – job. Ten provádí vlastní aplikaci jednoho skriptu nebo několika skriptů (podle typu jobu) na jeden až nekonečně mnoho serverů. To je to, co činí tento nástroj opravdovým automatizačním prostředkem. Díky přítomnosti RSCD agentů je možno provést doručení skriptu na server, tam vykonat kód skriptu a převzít jeho výsledky. Doslova jedním „kliknutím myši“ lze hotový a odladěný skript aplikovat. Tímto způsobem to lze provést na několika tisících serverech současně nebo postupně. Efekt takového přístupu je zřejmý, dojde k významnému ušetření času technika při aplikaci skriptu.

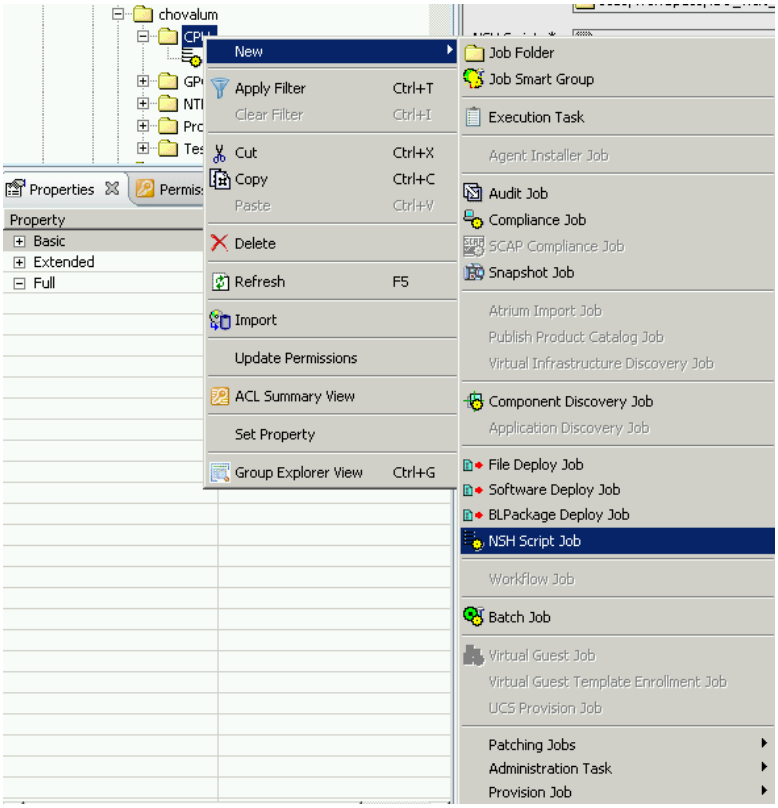
Joby slouží jako řídící objekt, provádějící sled událostí nakonfigurovaných v tomto jobu postupně tak, jak je požadováno. V tomto případě joby doručují vykonavatelný kód vlastního NSH skriptu provádějící změnu obsahu konfiguračního souboru PATROL Agenta na serveru. NSH skript sám o sobě tedy neví, kde bude jeho kód vykonán. To je právě výhradně určeno jobem. NSH skript bez příslušného jobu není vykonatelný. Job musí proto existovat vždy.

Toto tvrzení platí i naopak. Job se vždy odkazuje na již existující kód skriptu. Postup vývoje jobu kopíruje vývoj skriptu, respektive oba úkoly jsou spolu úzce provázány. Na počátku vývoje, kdy je potřeba odzkoušet a doladit jednotlivé části NSH skriptu, je potřeba mít vytvořeny i testovací joby. Ty aplikují testovací části NSH skriptů na testovací servery. Požadavky na testovací joby budou mít jiný charakter než požadavky na finalizovanou verzi jobu. V průběhu vývoje a ladění kódu skriptu je potřeba mít daleko větší přehled o jednotlivých fázích vykonávání skriptu. Typicky se provádí výpisy všech událostí po vykování kódu každé související sekce skriptu. V okamžiku ukončení vývoje je důležitý pouze konečný verdikt úspěšnosti provedení celého skriptu. Během vývoje skriptu jsou k dispozici i statusy, které poskytuje přímo prostředí BSA. Jsou to exit kódy o úspěšnosti provedení jobu a přehledná historie všech provedených testů a jejich výsledků.

5.1 Vytvoření automatizačního jobu

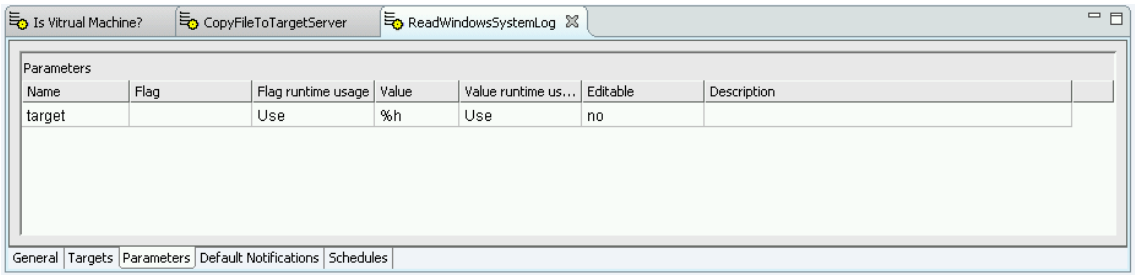
Vlastní job musí být vytvořen podobně jako NSH skript v příslušné složce na aplikačním serveru prostředí BSA. Existuje předdefinovaná sada jobů pro různá využití. Například Execution Task je job mající za úkol postupně vykonat několik samostatných NSH skriptů. File Deploy Job je job pro nakopírování konkrétního souboru mezi servery. Patching job je job pro provedení „patchování“ serveru. Všechny typy úloh mají stejnou logiku a teoreticky je možno každou úlohu provést za pomoci jednoho univerzálního typu, a to NSH script jobu.

Pro nastavení monitoringu byl použit základní typ, tedy NSH script job. Tento typ jobu byl vytvořen v příslušné složce na aplikačním serveru BSA, jak je patrné na obrázku 12.



Obrázek 12: Vytvoření NSH Script Jobu

Na obrázku 13 je pak vidět okno, ve kterém je zadán vstupní parametr, který je v tomto případě názvem serveru.



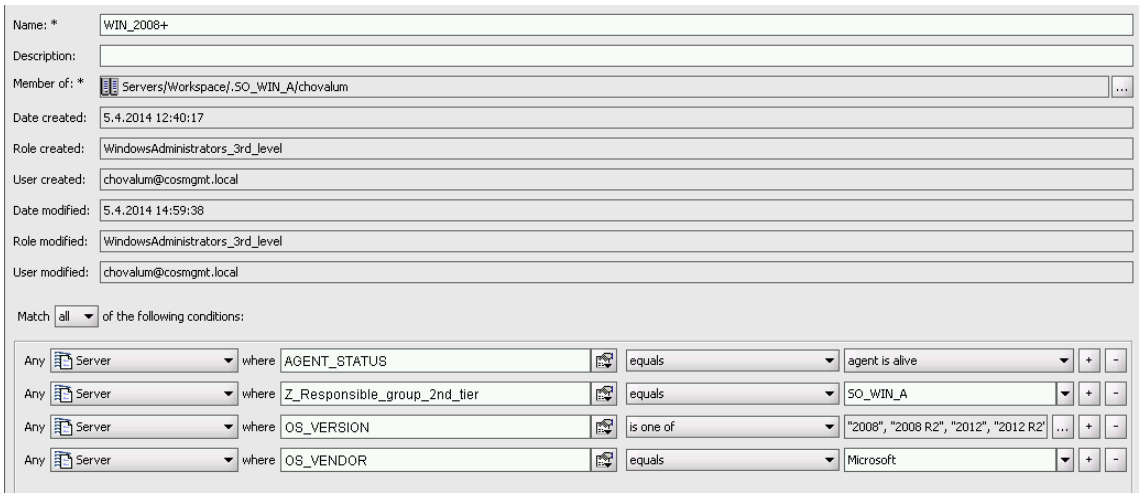
Obrázek 13: NSH Script Job, předání parametru

Pokud je to možné, jsou nastavovací skripty volány sériově, tedy po dokončení konfigurace jednoho serveru se zpustí konfigurace dalšího. V tomto případě je lépe zajištěna možnost logování případných návratových hodnot skriptu. Tak je zajištěna kontrola provedení skriptu na jednotlivých serverech. Nicméně, tento postup byl použit pouze v části vývoje a ladění. Pro vlastní aplikaci, kdy se toto nastavení aplikuje na tisíce serverů a může dojít k zablokování

provedení jednoho skriptu i na několik hodin (a to i při nastavení time – outu na provedení skriptu). Pro provedení konečné změny konfigurace nastavení monitoringu byl využit paralelní proces spouštění jobu za pomoci volby „Execute the script separatelly againtst each host“ (viz obrázek 11, kapitola 4.2). Tato volba zajistí souběžné spuštění na všechny servery. Toto spuštění lze provádět opakovaně tak dlouho, dokud nebude na všech dostupných serverech jeho provedení úspěšné.

5.2 Spuštění automatizačního jobu

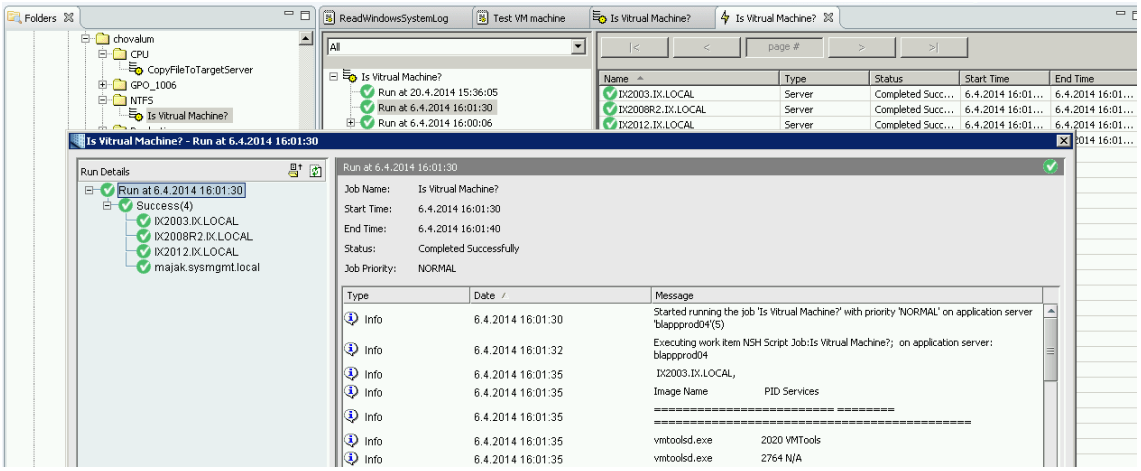
Aby bylo možno provést nastavení nové konfigurace, musí se definovat cílová skupinu serverů, na kterých je změna vyžadována. I zde poskytuje prostředí BSA významnou pomoc. Na výběr je několik možností, jak takovou skupinu definovat. Jako první se nabízí logická možnost definovat ji výčtem serverů. Tato možnost ovšem přichází v úvahu stejně jako ruční konfigurace v případě jednotek serverů. Ruční výběr serveru za serverem metodou step by step je samozřejmě zdlouhavý a nezabrání chybám ve výběru. Ani možnost načtení tohoto seznamu ze vstupního souboru dat není o mnoho lepší. Seznam by musel být nějakým způsobem vygenerován, jeho ruční tvorba nenabízí žádnou automatizaci. BSA však nabízí něco, co je možno považovat za klíčovou výhodu automatizace. Jsou to tzv. „Smart Groups“, tedy inteligentní, dynamicky vytvářené skupiny serverů. To, jaké servery se do této skupiny načtou, je možno ovlivnit za pomoci desítek atributů, které jsou k dispozici automaticky. BSA totiž udržuje obsáhlý depozit všech parametrů serverů a dynamicky je aktualizuje. Stačí zvolit správné atributy a příslušnou skupinu vhodně pojmenovat. Příklad výběru atributů pro vytvoření smart skupiny obsahující všechny servery s operačním systémem verze Windows 2008 a vyšší je na obrázku 14.



Obrázek 14: Smart group pro servery Windows 2008+

5.3 Výsledky aplikace automatizačního jobu

Nyní, když je definována cílová skupina serverů, existuje odladěný skript, může být vytvořen příslušný job. Následně nezbývá než tento job aplikovat na definovanou skupinu serverů.

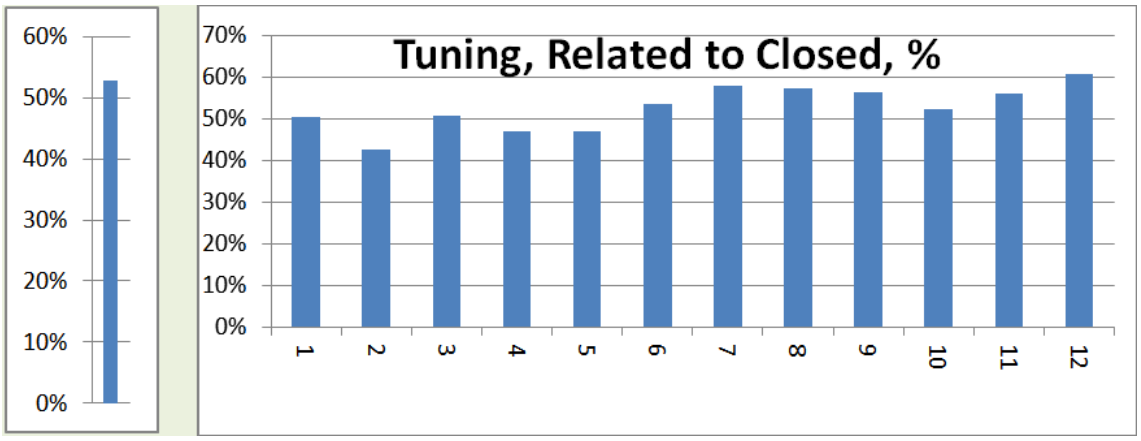


Obrázek 15: Výsledek provedení jobu

Ukázka, jak vypadá výstup provedení tohoto jobu, je na obrázku 15. Je vidět celou historii, kdy byl daný job aplikován, na jaké servery byl aplikován a po „rozkliknutí“ příslušného data pak detailní výsledky aplikace na konkrétní servery. Podle typu jobu jsou zobrazeny výsledky buď souhrnně pro celou skupinu serverů, a nebo se servery automaticky sloučí do skupin podle toho, zda byl skript proveden úspěšně či nikoliv. U neúspěšně provedených skriptů pak BSA poskytuje informaci o důvodech neprovedení jobu (tato informace není na obrázku 15 k dispozici, všechny skripty byly provedeny úspěšně).

6 NASAZENÍ AUTOMATIZACE, KONEČNÉ VÝSLEDKY

Myšlenka nasazení automatizace právě v oblasti nastavení monitoringu pramení z výsledků analýzy problem managementu v naší firmě. Právě tento proces je klíčovým pro aplikaci tzv. „continual service improovementu“, nekončícího cyklu zlepšování kvality poskytované služby. Výstupem Root Cause analýzy všech řešených problémů monitoringu bylo zjištěno, že ve více než 50 % případů je řešením právě požadavek na následnou úpravu monitoringu (tuning – vyladění, viz obrázek 16). Tedy nastavení monitorovaných parametrů tak, aby nedocházelo ke zbytečnému zatěžování specialistů řešením zbytečných varování na základě špatně nastaveného monitoringu.



Obrázek 16: Procentuální podíl požadavků na úpravu monitoringu v průběhu jednoho roku

Zároveň je velmi často požadováno, aby tyto úpravy byly provedeny na skupině serverů současně a stejně (nikoliv pouze jednotlivě). Dosud však díky absenci příslušného programového vybavení nebylo možno naplnit představu o automatizaci takového nastavení. Na základě strategického plánu firmy byl zakoupen automatizační nástroj firmy BMC, stejného výrobce, který dosud poskytuje firemní řešení pro infrastrukturu monitoringu. To umožnilo velmi úzké propojení mezi jednotlivými systémy.

Očekávání po nasazení tohoto sofistikovaného nástroje jsou stále vysoká a naplňují se zatím postupně. Automatizace nastavení monitoringu je jen jednou z mnoha možností uplatnění tohoto nástroje. Díky dostatečně kvalitním podkladům a provedeným analýzám, jejichž souhrn byl uveden v kapitole 3, bylo možno předpokládat, že dojde k úsporám především v náročnosti nastavování jednotlivých parametrů. Prioritní byla myšlenka vytvoření takových jobů, které by bylo možné použít pro nastavení konkrétních parametrů na konkrétních serverech. Cílem bylo především odstranění chyb a nekonzistenci nastavení monitoringu, čímž se prioritně mělo předejít spíše fatálním výpadkům monitoringu, než ušetřením času techniků zabývajících se kontrolou stavu serverů v případě příchodu nadbytečné informace z monitoringu.

Výsledky v jednotlivých kategoriích jsou založeny na analýze všech vygenerovaných incidentů a jejich porovnáním za období před a po provedení nastavení monitoringu. Vyhodnocení bylo provedeno na firemních datech, které jsou pro účely této práce k dispozici anonymizované ve formě tabulky CSV (příloha 1 až 3). Jedná se tedy o naprosto přesný způsob vyhodnocení. Přímo nad těmito daty se za pomoci kontingenčních tabulek prováděly souhrny dle sledovaného období. V této práci jsou výsledky kvůli přehlednosti převedeny do formy grafů.

6.1 Výsledky změny nastavení monitoringu CPU

S dostatečným odstupem po provedení změny nastavení monitoringu proběhlo vyhodnocení. Jeho výsledky prokazují výrazné snížení nadbytečných incidentů CPU. K tomuto efektu mohlo dojít pouze za předpokladu kvalitně provedené analýzy vstupních dat. Původním cílem bylo zabránit zejména velmi náročnému postupu v nastavování. Jen v menší míře se předpokládala úspora času věnovaná kontrolám stavu serveru, ke kterým docházelo typicky již při příjmu tiketu do systému. Standardně docházelo ke kontrole přetížení CPU do jedné hodiny po vygenerování tiketu a zde nastaly dvě možné situace:

- stav přetížení již pominul a CD specialista tiket uzavírá jako dočasný problém,
- stav přetížení trvá a tiket je přesunut na další vrstvy k analýze příčin přetížení. Na dalších vrstvách však došlo ke zpoždění v délce trvání další hodiny. Při opětovné kontrole specialistou již byla opět většina incidentů uzavřena jako dočasný problém způsobený spuštěním zálohovacího jobu. Toto se navíc dělo v nočních hodinách, kdy jsou směny obsazeny výrazně menším množstvím specialistů než během dne.

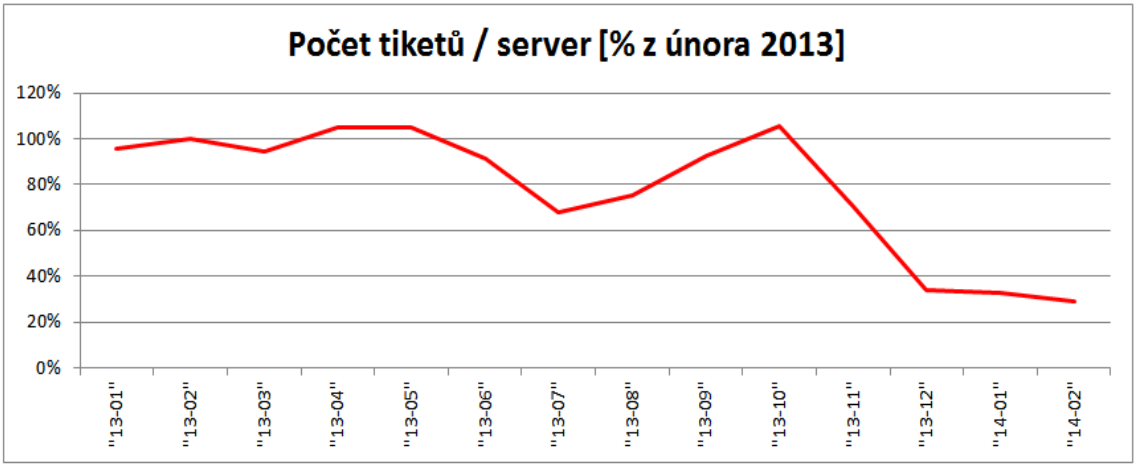
Po aplikování nastavovacích jobů došlo k významné redukci počtu incidentů na jeden server. Toto je poměrové měřítko nutné k provedení objektivity měření. Počet serverů v delším časovém období se může výrazně měnit (nejčastěji se zvyšuje). Původní očekávání se pohybovala na úrovni 50% redukce, neboť skript nešlo aplikovat na 100 % monitorovaných serverů. Zejména panovaly obavy z možných problémů při zanedbání monitoringu kritických serverů. Proto na těchto serverech nemohlo být takové nastavení aplikováno.

Jak je vidět v tabulce 5, došlo k poklesu až na hodnotu 29 % původního průměrného počtu incidentů. Z výsledků je tedy patrné významné snížení absolutní hodnoty počtu automatických incidentů (obrázek 17, červená křivka) vůči zvolenému referenčnímu měsíci 2-2013. Tento měsíc byl zvolen jako reprezentant průměrné roční hodnoty před provedením konfiguračního jobu. Došlo tedy k reálné redukci o 70 %, která se potvrdila i po dvou měsících od aplikace.

Rok	2013	2013	2013	2013	2013	2013	2013	2013	2013	2013	2013	2013	2014	2014
Měsíc	1	2	3	4	5	6	7	8	9	10	11	12	1	2
Control Desk	773	973	959	1125	1097	930	791	743	1155	979	822	326	127	91
Windows specialisté	714	665	648	659	601	431	395	563	533	846	491	299	483	431
UNIX specialisté	84	101	121	111	90	99	91	99	75	128	144	51	22	13
Aplikační specialisté	322	240	179	242	343	438	138	177	194	294	74	68	79	95
Součet	1893	1979	1907	2137	2131	1898	1415	1582	1957	2247	1531	744	711	630
Počet [% z února]	96%	100%	95%	105%	105%	91%	68%	75%	93%	106%	70%	34%	33%	29%
Specialisté na 2 a 3 level [%]	59%	51%	50%	47%	49%	51%	44%	53%	41%	56%	46%	56%	82%	86%

Tabulka 5: Konkrétní počty incidentů pro monitoring CPU za rok 2013

Skripty byly převážně nasazeny v období říjen, listopad a prosinec 2013. Z křivky na obrázku 17 je patrný vývoj počtu incidentů za celý rok 2013. Z jeho průběhu lze dále vysledovat roční průběh počtu přetížení CPU v celé infrastruktuře. Je viditelný pokles v období letních prázdnin (sezónní snížené využívání serverů) a naopak silný nárůst v podzimních měsících (rovněž typické sezónní vytěžování serverů v produkci) ukončený až nasazením nového nastavení. K viditelnému snížení by bez změny nastavení monitoringu nedošlo.



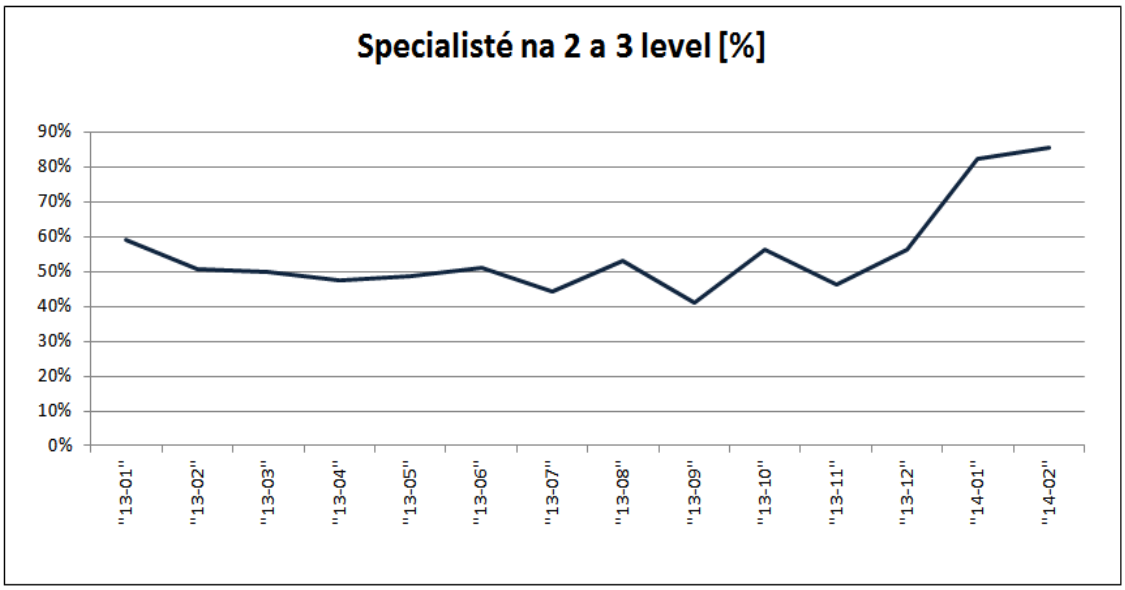
Obrázek 17: Křivka zachycující vývoj počtu tiketů na jeden server za rok 2013

Toto významné snížení si lze vysvětlit především tím, že se nyní nemonitorují přetížení způsobená typickými nočními zálohovacími úlohami. Naproti tomu reálné problémy přetížení CPU monitorovány zůstávají a jsou předávány k řešení specialistům. Ti mají nyní více času na vyhledání skutečné příčiny tohoto přetěžování CPU. Tím pádem dochází k dalšímu pomalému snižování počtu přetížení CPU, což je z grafu také patrné (období 13/12...14/02). Tento efekt byl neočekávaný a je vnímán rovněž pozitivně.

Na obrázku 18 je patrný trend přesunu práce z první úrovně specialistů (Control Desk) na druhou a třetí úroveň specialistů (souhrnně za Windows, Unix a aplikační vrstvu) věnujícím se hledání skutečné příčiny problému přetěžování CPU (jsou použita data z tabulky 5). Důležitá je

informace, že i přes redukci počtu automatických incidentů nedošlo ve sledovaném období ani k jednomu výpadku poskytovaných služeb, jejichž příčinou by bylo přetížení procesoru.

Reálné výsledky předčily očekávání a dostavilo se téměř úplné odstranění „zbytečné práce“ s kontrolou stavu serverů v situacích, kdy nehrozí reálný problém. Vyhodnotit velikost úspory pracovní síly při nastavování jednotlivých serverů je obtížné. Nejsou totiž k dispozici průkazná data o reálné potřebě množství práce nutné pro provádění těchto změn, a to ani před a ani po provedeném automatickém nastavení monitoringu CPU.



Obrázek 18: Poměr řešených tiketů podle úrovně specialistů

6.2 Monitoring NTFS a Group Policy

Změny nastavení monitoringu NTFS a GPO byly provedeny až v průběhu roku 2014. Dostatečné množství dat pro definitivní posouzení vlivu nového nastavení není k dispozici. Pro predikci efektivity nového nastavení musela být využity data pouze do konce měsíce dubna. Vyhodnocení průměrných dat za první čtvrtletí roku 2014 a období po nasazení nového nastavení ukazuje tabulka 6.

NTFS	leden...březen	duben	zůstane	Redukce
naměřené	1320	174		
roční predikce	5353	2113	39%	61%

Tabulka 6: Výsledek nastavení monitoringu pro NTFS

Z ní vyplývá, že i v případě NTFS bude velmi reálně dosaženo redukce mezi 60-70 % událostí. Oproti předpokladům se i zde podařilo dosáhnout vyšší míry snížení počtu incidentů. Dále byl vytvořen testovací job, který po spuštění na server podá specialistovi informaci, zda událost vygeneroval VHD nebo fyzický disk. To umožňuje specialistům spolehlivě určit příčinu bez nutnosti se na server přihlašovat a hledat příčinu na serveru. Reálně se jedná o úsporu času věnovanou každému jednotlivému tiketu. Dle přílohy 7 lze usuzovat, že se jedná o 8 tiketů denně, které ušetří technikům asi cca 25 minut práce. Úspora se jeví jako nevýznamná a zejména obtížně prokazatelná. Prokazatelný je ovšem fakt, že tato kontrola je jednoznačná a tímto způsobem lze předejít chybám v úsudku specialisty, zda se o chybu jedná či nikoliv.

Redukci událostí GPO za stejné období do konce dubna eviduje tabulka 7. Míra snížení je ve výši 47 %. Výsledek nastavení přesně odpovídá vstupním předpokladům. V další fázi bude vyvinut automatizační job, který bude reagovat na výskyt události aktivním způsobem. Bude ověřovat identitu přihlášeného uživatele a reagovat na nastalou situaci podle příslušnosti uživatele v doménových skupinách.

GPO	leden...březen	Duben	zůstane	Redukce
naměřené	1729	303		
roční predikce	7012	3682	53%	47%

Tabulka 7: Výsledek nastavení monitoringu pro GPO

6.3 Vyhodnocení možností dalšího rozšíření

Ve všech oblastech, kde byla použita některá z metod automatizace, bylo dosaženo významných výsledků. Výsledek Bussines analýzy pro výběr nejvhodnější varianty z hlediska náročnosti provedení skriptu a jeho nasazení není v době uzavření této bakalářské práce znám. Není tudíž jasné, jakou metodou se bude ubírat další vývoj nových jobů. Zda se bude i nadále optimalizovat monitoring a tím snižovat požadavky na celou infrastrukturu monitoringu a BSA. A nebo se půjde cestou maximalistického zpracovávání všech událostí a následně analýzy těchto událostí prostřednictvím automatizačních tasků. V tomto případě je však nutná podpora dalších teamů z oblasti správy monitoringu. Automatizované spuštění totiž vyžaduje naprogramování speciálního rozhraní pro předávání parametrů ze zachyceného incidentu do jobu a následně NSH skriptu a samozřejmě i nazpět.

Nastavení monitoringu CPU pomocí skriptu bylo pilotním projektem v rámci naší firmy, který jsem osobně organizoval. Jeho provedení nebylo nijak jednoduché i přes to, že se jednalo o velmi jednoduché nastavení. Bylo ovšem nutno přesvědčit management a nadchnout jednotlivce pro tuto práci. Během studia na této bakalářské práci se automatizace v naší firmě právě rozbíhala a v tuto chvíli již běží desítky projektů současně. Mnohé z nich probíhají paralelně. Implementace probíhá za plného provozu a není věnován dostatek prostoru pro nalézání optimální cesty k využití BSA. Většina specialistů nachází vlastní způsoby použití.

Základní principy však fungují a základní cesty jsou již prošlapány a postupně objevovány dalšími a dalšími teamy. Dochází k proškolování desítek techniků pro prostředí BSA. Mnoho specialistů vytvořilo alespoň několik skriptů, které mají uloženo ve svých „workspace“. Mnoho skriptů je sdíleno mezi jednotlivci či skupinami specialistů na daných platformách.

6.4 Další vývoj, metodika výběru oblastí pro nasazení

Praktické využití poznatků z této bakalářské práce lze spatřovat v prvé řadě ve výběru vhodné metodiky pro posuzování toho, jak najít ty případy, kdy je nasazení automatizace monitoringu nejefektivnější. Dovodit lze, že hlavním hlediskem je kritérium maximálního využití. Tedy případy, kdy je provedení skriptu jednoduché a počet nasazení největší. Další v pořadí je největší úspora manuálních opakujících se a relativně komplikovaných úloh. Vytvoření skriptu pro takové úlohy již tak jednoduché není, ale při ručním nastavování existuje riziko lidské chyby. Na posledním místě jsou složité skripty aplikované na velmi omezeném množství serverů. Zde je efekt nejnižší a v podstatě nemá v praxi význam takové skripty vytvářet. Alespoň ne v počátečních stádiích implementace automatizace, kdy prioritní dvě kategorie produkují dostatek materiálu pro automatizaci.

6.5 Využití pro praxi

Pro různá prostředí se bude optimální metoda automatizace vždy lišit a bude záviset mimo jiné i na finančních a technických možnostech konkrétní firmy. V rámci této bakalářské práce lze hodnotit pouze metody použité přímo na pracovišti. Tedy produkty BSA, kde se jednalo o metody přímé změny nastavení monitorovaných parametrů u změny nastavení monitoringu CPU. Tato cesta se ověřila a je možné v ní pokračovat. Pravděpodobně dojde k významnému rozšíření jednoúčelových jobů umožňujících rychlé a spolehlivé nastavení různých parametrů monitoringu. Jejich výběr bude zcela určitě odpovídat reálným potřebám specialistů. Výstupem pravděpodobně budou i nadále Root Cause analýzy vzešlé z problem managementu.

Další možnosti, které lze automatizovat jsou ve využití BSA Orchestratoru. To je nástroj, který umožní reagovat na vznik události spuštěním jobu, který pak zajistí automatickou korekci chybného stavu. Tato varianta řešení automatizace nesouvisí s předmětem této práce, tedy automatizace nastavení monitoringu.

7 ZÁVĚR

Byly vytvořeny tři NSH skripty a prostřednictvím jobů bylo provedeno automatické nastavení monitoringu serverové infrastruktury. Prokázal se, že nasazením automatizace monitoringu došlo k významné redukci počtu zaznamenaných událostí u monitorovaných parametrů. Oproti původním teoretickým předpokladům bylo dosaženo o 20 % většího efektu. Navíc se zajistila konzistence nastavení monitoringu a snížila se pracnost při nastavování jednotlivých parametrů. Došlo ke zvýšení kvality poskytovaných služeb a snížení náročnosti a chybovosti při implementaci změn nastavení monitoringu serverové infrastruktury.

V našem firemním prostředí se jednalo o vůbec první nasazení unifikovaného nastavení na všechny servery a reálně hrozila možnost, že provedená analýza a aplikované nastavení nebude odpovídat předpokladům. Dosavadní praxe ruční konfigurace jednotlivých serverů totiž za dlouhá léta vytvořila značně nekonzistentní prostředí, ze kterého bylo obtížné zajistit relevantní data. Další hrozbou byl výběr příliš striktního nastavení a možnosti nezachycení reálných problému včas s následkem výpadku poskytované služby zákazníkovi. Ani jedno z těchto rizik se neprojevilo. Naopak se uplatnila výhoda rekonfigurace špatně nastavených PATROL Agentů, kde vinou lidského faktoru byly některé parametry monitorovány špatně nebo vůbec (zejména u nastavení monitoringu NTFS). Problémy se vyskytly pouze při aplikaci nastavení na jednotlivých PATROL Agentech z důvodu velké roztříštěnosti verzí. Vlastní nastavovací NSH skripty byly vyvinuty v poměrně krátké době a jejich funkčnost je díky jednoduchosti přístupu v prostředí BSA velmi spolehlivá. Implementace odladěného skriptu je díky logice spouštění jobů nad skupinami serverů velmi jednoduchá a rychlá. Logika jobů je to, co činí největší výhodu automatizace. V případě jakýchkoliv problémů s aplikací nastavení existuje velmi přehledná kontrola aktuálního stavu, kde byl skript aplikován úspěšně, kde ne a jaké jsou důvody neprovedení skriptu. Nejčastější příčinou je nedostupnost RSCD agenta z důvodu aktuální nedostupnosti serveru (síťový problém) nebo aktuálního vytížení serveru. Další velká skupina problémů pak souvisí s nutností restartovat PATROL Agenta pro aplikaci nového nastavení, což s sebou nese riziko výpadku monitoringu, pokud PATROL Agent nenaběhne po restartu korektně. Řešení této části problémů není předmětem této práce.

Další vývoj nasazení automatizace nastavením monitoringu je velkou výzvou pro všechny techniky, aby dodali relevantní analýzy pro většinu důležitých monitorovaných parametrů a umožnili tak plné využití potenciálu tohoto jednoduchého a spolehlivého přístupu pro nastavení i těch nejkomplikovanějších parametrů monitoringu. V současné době probíhá intenzivní školení všech techniků pro prostředí BSA. Dále jsou iniciovány skupiny architektů jednotlivých systémů tak, aby dodali technickou analýzu stavu monitoringu a návrhy na zlepšení, která lze hromadně aplikovat. Další možnosti, které lze automatizovat jsou ve využití BSA Orchestratoru. To je nástroj, který umožní reagovat na vznik události spuštěním jobu, který pak zajistí automatickou korekci chybného stavu. Nicméně, v praxi je vždy na prvním místě efektivita a potřeba se soustředit pouze na taková řešení, která přinášejí nejvyšší efektivitu. Lidská práce je sice obecně dražší než práce stroje, ale vytvoření „dokonalého“ softwaru či samoopravného skriptu je časově natolik náročné, že tato varianta není vždy optimální.

LITERATURA

- BMC SOFTWARE. *BMC BladeLogic Network Shell Command: Reference manual*. Version 8.1.00. HOUSTON, USA, 2011.
- STOLZ, Annette. *Microsoft Windows Server 2003 skripty: velká kniha řešení*. Vyd. 1. Překlad David Čepička. Brno: Computer Press, 2007. 699 s. ISBN 978-80-251-1668-5.
- SHAH, Steve. *Administrace systému Linux: překlad čtvrtého vydání*. 1. vyd. Praha: Grada, 2007. 426 s. ISBN 978-80-247-1694-7.
- KOLEKTIV AUTORŮ. *Automatizace a automatizační technika 1: systémové pojetí automatizace*. 1. vyd. Brno: Computer Press, 2012. 217 s. ISBN 978-80-251-3628-7.
- BUCKSTEEG, Martin. *ITIL 2011*. 1. vyd. Brno: Computer Press, 2012. 216 s. ISBN 978-80-251-3732-1.
- Microsoft. *Resources and Tools for IT Professionals | TechNet* [online]. 2013 [cit. 2013-11-22]. Dostupné z: http://technet.microsoft.com/
- JONES, Don. *Automatizace správy a skriptování Microsoft Windows*. Vyd. 1. Překlad Jan Gregor, Libor Král. Brno: Computer Press, 2006. 404 s. ISBN 80-251-1261-6
- VMware. VMware KB [online]. 2014-01-15 [cit. 2014-02-04]. Dostupné z: http://kb.vmware.com/selfservice/microsites/microsite.do

SEZNAM PŘÍLOH

1) Analýza incidentů CPU, csv
2) Analýza incidentů NTFS, csv
3) Analýza incidentů GPO, csv
4) Zdrojové kódy NSH skriptů, neveřejná CD příloha